Accepted Manuscript

Capturing Relative Motion and Finding Modes for Action Recognition in the Wild

Olusegun Oshin, Andrew Gilbert, Richard Bowden

PII: S1077-3142(14)00084-8
DOI: <http://dx.doi.org/10.1016/j.cviu.2014.04.005>
Reference: YCVIU 2127

To appear in: *Computer Vision and Image Understanding*

Received Date: 12 February 2013
Accepted Date: 5 April 2014

Please cite this article as: O. Oshin, A. Gilbert, R. Bowden, Capturing Relative Motion and Finding Modes for Action Recognition in the Wild, *Computer Vision and Image Understanding* (2014), doi: <http://dx.doi.org/10.1016/j.cviu.2014.04.005>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Capturing Relative Motion and Finding Modes for Action Recognition in the Wild

Olusegun Oshin, Andrew Gilbert*, Richard Bowden*

*Centre for Vision, Speech and Signal Processing, University of Surrey,
Guildford, Surrey GU2 7XH United Kingdom*

Abstract

“Actions in the wild” is the term given to examples of human motion that are performed in natural settings, such as those harvested from movies [1] or Internet databases [2]. This paper presents an approach to the categorisation of such activity in video, which is based solely on the relative distribution of spatio-temporal interest points. Presenting the Relative Motion Descriptor, we show that the distribution of interest points alone (without explicitly encoding their neighbourhoods) effectively describes actions. Furthermore, given the huge variability of examples within action classes in natural settings, we propose to further improve recognition by automatically detecting outliers, and breaking complex action categories into multiple modes. This is achieved using a variant of Random Sampling Consensus (RANSAC), which identifies and separates the modes. We employ a novel reweighting scheme within the RANSAC procedure to iteratively reweight training examples, ensuring their inclusion in the final classification model. We demonstrate state-of-the-art performance on five human action datasets.

Keywords: Action Recognition, Relative Motion Descriptor, Mode Finding, Outlier Detection, RANSAC

[☆]This work is supported by the EU FP7 Project Dicta-Sign (FP7/2007-2013) under grant agreement no 231135, and the EPSRC project Making Sense (EP/H023135/1).

*Corresponding author

Email addresses: o.oshin@surrey.ac.uk (Olusegun Oshin),
a.gilbert@surrey.ac.uk (Andrew Gilbert), r.bowden@surrey.ac.uk (Richard Bowden)

1. Introduction

Human action recognition from video has gained significant attention in the field of Computer Vision. The ability to automatically recognise actions is important because of potential applications in video indexing and search, sports analysis, activity monitoring for surveillance, assisted living purposes, etc. Excellent results have been obtained on simulated actions in simplified settings. However, natural actions in uncontrolled environments, such as movies and personal video collections have proven more difficult. The task is especially challenging due to variations in illumination, scale, camera motion, viewpoint, background, occlusion, action length, subject appearance and style.

In order to adequately describe actions in the presence of the above variations, recent recognition approaches make use of a combination of feature types. They often combine shape and motion, or include contextual information, while others employ object detectors to bolster recognition confidence. This can cause problems as many of the challenges stated in the paragraph above are appearance based. We are motivated to just use motion, inspired by Johansson’s [3] human perception experiments using Point Light Displays. These experiments show that it is possible to observe detailed properties of human actions and actors based entirely on the dynamics of their motion. Hence, given a set of point-lights that appear randomly placed to human observers, the introduction of their motion can clearly convey what action is being performed. This motivates our approach. This paper therefore investigates the sole use of dynamics for automatic action recognition in complex videos using interest point positions without their individual appearance information.

An action can be defined as a collection of atomic events at various spatial and temporal regions of a video. Our method is based on the premise that those events are better defined by motion information, given that the appearance of subjects and the background are subject to change. It is assumed that the motion information provides a discriminative representation. A novel representation of actions in video is therefore proposed, which captures the relative distribution of motion-based interest points by encoding their local spatio-temporal configuration in an efficient manner. This results in an action descriptor, which, in vectorised form, can be learnt using a discriminative classifier. In contrast to other interest point distribution-based representations, which assign labels to points based on the appearance of

their local spatio-temporal neighbourhoods, the proposed representation ignores such labels, and relies on the strengths of responses within less localised regions of the video.

Furthermore, approaches to action recognition typically attempt to generalise over all examples from the training data. Given the amount of variability within actions classes in natural settings, it is considered unrealistic to assume that all aspects of variability can be modelled by a single classifier. This paper presents an approach which tackles variability in complex action examples by assuming the presence of noisy examples, which cause non-separability between classes; and by assuming that there exist multiple modes within the set of examples of any one class. This is necessary as there are no constraints placed on actions performed and captured in movies and personal videos. Figure 1 shows examples of the action categories, *Get Out of Car* and *Hand Shake* from the Hollywood2 dataset [1]. It is clear from these examples that, while the same semantic action is being performed, all examples appear radically different in appearance which leads to multiple modes within the same action. Despite these variations, the examples are given one semantic label, and analysis of these examples as a single group may limit classification performance.

Figure 1: Examples of two actions of the Hollywood2 dataset [1], all showing the different modes of the same action.

It can be seen in Figure 1 that category labels can be broken into sets of different class subsets depending on the placement of the camera with respect to the car, or the individuals shaking hands, for *Get Out of Car* and *Hand Shake* respectively. Therefore, instead of considering all examples of an action category label as one class, we analyse examples to determine inherent *modes* or groups. An action class can, therefore, be partitioned based on these groups, significantly simplifying the training and classification task.

We achieve this by employing a variant of the Random Sampling Consensus (RANSAC) algorithm [4] to training examples of a class. Once the modes of an action class are found, several sub classifiers are created, which are applied independently to unseen examples. Action categories can therefore be split into subsets of consistent modes, which cover variability of action, environment and viewpoint. While extensive works exist on local classification methods for object category recognition [5, 6], human pose estimation [7], etc. [8], the assumption of multi-modality has not been explicitly applied to action recognition.

The layout for the remainder of this paper is as follows: Section 2 discusses related research. In Sections 3 and 4, we present the Relative Motion Descriptor and Automatic Mode Finding methods respectively. We describe our experimental setup in Section 5 and present recognition results in Section 6. Finally, conclusions are presented in Section 7.

2. Related Work

There is a considerable body of work exploring the description of actions in video for recognition. Many of the approaches make use of a sparse set of local interest points generated by the action [9, 10, 11, 12, 13, 14], and have demonstrated remarkable performance. Interest points are highlighted as salient regions based on response functions applied to the video, and eliminate the need for motion tracking and background subtraction, which are costly pre-processing steps. Examples include Laptev and Lindeberg’s [15] temporal extension of Harris 2D corners, Willems et al.’s [16] use of the determinant of the generalised 3D Hessian matrix, and Dollar et al.’s [11] separable linear filters.

Several descriptors have been proposed to encode the spatio-temporal support region of these interest points. These include local jet descriptors [9], vector of concatenated pixel gradients [11], generalisation of the SIFT and SURF descriptors [17, 18, 16], and the HOG/HOF descriptors [12]. The approach by Bregonzio et al [19] differs significantly from the existing interest point based representation in that only the global distribution information of interest points is exploited. The detected interest points are typically then used in a discriminative [9, 11] or generative [10] model.

While earlier action recognition methods were evaluated on simulated actions in simplified settings, more recent work has shifted focus to natural actions in unconstrained scenarios, *e.g.* personal video collections and movies.

As a result of this increase in complexity, recent approaches [20, 12, 1] make use of a combination of feature types. Laptev and Perez [20], distinguish between actions of Smoking and Drinking in movies, combining an optical flow-based classifier with a separately learned space-time classifier applied to a key frame of the action. The works of [12] and [1] recognise a wider range of actions in movies using concatenated HoG and HoF descriptors in a bag-of-features model, with [1] including static appearance to learn contextual information. Han et al. [21] capture scene context by employing object detectors and introduce bag-of-detectors, encoding the structural relationships between object parts. Similarly, Reddy and Shah [22] argue that as datasets increase in size, the use of scene context increases in importance. Whereas Ullah et al. [23] combine non-local cues of person detection, motion-based segmentation, static action detection, and object detection with local features. Liu et al. [13] also combines local motion and static features and recognise actions in videos obtained from the web and personal video collections.

In contrast to these multiple-feature approaches, our method makes use of one feature type. Then, instead of seeking to learn generalisation over all class examples, we argue that a single action can be split into subsets, which cover the variability of action, environment and viewpoint. Klipper-Gross et al [24] focus on motion encoding and on capturing local changes in motion directions.

Zhang et al [5] and Malisiewicz et al [6] both perform local classification of object category recognition, Zhang uses a k-Nearest Neighbour method to approximate the SVM, while Malisiewicz combine the effectiveness of a discriminative object detector with the explicit correspondence offered by a nearest-neighbour approach. Within video and action recognition a number of recent approaches have shown good performance by capturing the local spatio-temporal configuration of interest points. Gilbert *et al* [14] build compound hierarchical features based on the relationships of detected interest points, and use data mining to discover reoccurring patterns. Ryoo and Aggarwal [25] use a set of pairwise predicates to describe relationships between interest points, while Matikainen *et al* [26] build relative location probability maps of interest points. Kovashka *et al* [27] construct a hierarchy of vocabularies from spatio-temporal neighbourhoods of interest points, encoding the points and a configuration of their neighbours. Savarese *et al* [28] also captures the pairwise correlation of interest point labels based on their proximity. While these methods encode configurations based on the

appearance *and* location of interest points, our approach makes use of their locations and strengths only, discarding appearance information.

Recently, Guha and Ward [29] divided the video sequence into small sub-sequences. On the first frame of each sub-sequences they find the 2D interest points and calculate the different order moments of the pixels in the neighborhood of those points on each subsequent frame. They call it local motion pattern. They represent each video by sparse linear combination of STIP features.

3. The Relative Motion Descriptor

The aim of this work is to create a representation capable of discriminating between actions in complex scenes using action and scene dynamics as indicated by spatio-temporal interest points. In contrast to other methods tackling action recognition in complex scenes, we present a method that utilises the distribution of motion-induced features only. Actions can be described in terms of the motion of subjects and the objects with which they interact. A large portion of the motion may be localised with respect to the entire region of the subject, as is observed with a *Clapping* or *Answer Phone* action, while the motion of other actions are holistic, and may include background contextual motion, e.g. *Running*, *Horseback Riding*, etc. For the *Clapping* and *Answer Phone* actions, the response of interest point detectors will generally be greater along the arms than at other parts.

For a small spatio-temporal window within a video, it is assumed that the appearance of an interest point-inducing event remains relatively unchanged throughout the window. It follows that the relative response strengths of interest points associated with the motion within the region are assumed to exhibit minimal change. This eliminates the need for labelling or tracking of interest points, but requires a representation capable of capturing the characteristic motion conveyed by the distribution of response strengths. This method differs from prevalent appearance-based bag-of-features approaches such that, while those methods encode the appearance of interest point regions, ignoring their spatio-temporal correlation, our approach encodes spatio-temporal correlation, while ignoring interest point appearance.

Motion information is therefore captured implicitly from this correlation, making use of simple tests which measure relative response strengths of interest points. In contrast to Ryoo and Aggarwal [25], who construct histograms of particular pairwise relationships using a set of spatial (*near*, *xnear*, *ynear*,

far) and temporal (*equals*, *meets*, *before*, *overlaps*, etc.) predicates, this approach describes atomic events by random tests on the relative strengths of interest point responses between two *regions*, in a scanning window approach. This section describes the method in detail.

3.1. Descriptor Formulation

The proposed Relative Motion Descriptor (RMD) is inspired by the Randomised Ferns ensemble classifier [30], which are non-hierarchical classification structures derived from Random Forests. Each Randomised Fern consists of set of ordered binary tests, which split the space of data being classified.

Our approach makes use of these ordered tests. However, in contrast to Randomised Ferns, which are classifiers, we use the representation to create an action descriptor, which is then used in conjunction with a more powerful classifier, e.g. non-linear Support Vector Machines. The descriptor encodes the local spatio-temporal dependencies for action and scene motion in a scanning volume approach applied at all spatio-temporal locations within the video, and aggregates these dependencies in a vector for each action example.

A set of region tests $f_j, j = \{1, \dots, N_f\}$, are defined, where N_f is the number of region tests for a given volume. For a particular instance of a scanning volume i_{XYT} with dimensions X, Y, T taken from a video I , a region test is defined as a comparison of the amount of motion between two regions randomly positioned within the windowed volume at points x, y, t . These randomly positioned region comparison locations are then fixed over all scanning volume blocks at training and test. Given that the regions have spatial, σ and temporal, τ extents $\frac{X}{\sigma}, \frac{Y}{\sigma}, \frac{T}{\tau}$, and $\mathcal{M}_\nabla$ is the mean interest point response strength within a region r , the result of a binary node region test f_j can be given by,

$$f_j = \begin{cases} 1 & \text{if } \mathcal{M}_1 > \mathcal{M}_2; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The mean interest point response strengths, $\mathcal{M}_r$ is given by

$$\mathcal{M}(x, y, t, \frac{X}{\sigma}, \frac{Y}{\sigma}, \frac{T}{\tau}) = \frac{1}{OE} \sum_{x'=x}^{x+\frac{X}{\sigma}} \sum_{y'=y}^{y+\frac{Y}{\sigma}} \sum_{t'=t}^{t+\frac{T}{\tau}} \iota(x', y', t'). \quad (2)$$

where ι is the representation of the video in terms of the OE detected interest points, given by

$$\iota(x, y, t) = \begin{cases} \Re(I_{xyt}) & \text{if } \Re(I_{xyt}) > \mathbb{T}; \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

$\Re$ is the strength of the response function or confidence of the interest point detector. $\Re$ is obtained at point x, y, t in the video, and $\mathbb{T}$ is the threshold above which interest points are detected. $\mathbb{T}$ can be varied so that either dense or sparse interest points are detected.

The values of σ and τ are randomly uniformly generated for each region compared within the scanning volume. This enables the encoding of interest point distributions of various spatial and temporal extents with the volume.

For each scanning volume instance, arrays of node tests, $A_k, k = \{1 \dots N_A\}$ are uniformly randomly generated, where an array comprises of a specific set of node tests applied to scanning volume instances. Figure 2 illustrates the computation of the Relative Motion Descriptor, highlighting nodes within one of the arrays of node tests applied to a region within a scanning volume instance. The scanning volume responses, δ for one array of node tests over

Figure 2: Overview of the Relation Motion Descriptor, showing arrays of node tests applied to a spatio-temporal scanning volume instance obtained from the video.

the entire video sequence is modelled with a histogram of size n^{N_f} . Resulting histograms of responses for all N_A arrays over the sequence are concatenated, creating a $N_A \times n^{N_f}$ dimensional vector, where n is the radix of the node test, for example binary ($n = 2$) or ternary ($n = 3$). Figure 3 illustrates a number of arrays of node tests on one scanning volume, and depicts histograms obtained for each array for all scanning volume instances in a video. This concatenated vector of node test array responses accumulates the frequency of certain local motion patterns, creating a discriminative action representation.

Figure 3: Each array of node tests produces a histogram representing the responses obtained at the scanning volume over the entire video sequence. Histogram responses over the entire video for each array of node tests are concatenated. δ_{xyt} corresponds to the decimal representation of the n^{N_f} long word to the radix n

3.2. Histogram Computation

Results of node tests can be encoded using higher radices, e.g. ternary or quaternary, instead of binary. This allows for the encoding of additional local motion information, without increasing the number of node tests, and consequently the computational complexity of the method. To this end, equation 1 above can be modified such that the node test result, f_j , is given by,

$$f_j = \begin{cases} 2 & \text{if } (\mathcal{M}_1 - \mathcal{M}_2) \geq \hbar \times \max(\mathcal{M}_1, \mathcal{M}_2); \\ 1 & \text{if } (\mathcal{M}_1 - \mathcal{M}_2) \leq -\hbar \times \max(\mathcal{M}_1, \mathcal{M}_2); \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

for ternary tests, or

$$f_j = \begin{cases} 3 & \text{if } (\mathcal{M}_1 - \mathcal{M}_2) > \hbar \times \max(\mathcal{M}_1, \mathcal{M}_2); \\ 2 & \text{if } 0 < (\mathcal{M}_1 - \mathcal{M}_2) \leq \hbar \times \max(\mathcal{M}_1, \mathcal{M}_2); \\ 1 & \text{if } 0 \geq (\mathcal{M}_2 - \mathcal{M}_1) \geq -\hbar \times \max(\mathcal{M}_1, \mathcal{M}_2); \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

for quaternary tests. In Equation 1, the boundary that defines the result f_j , is 0, and the upper component of the equation can be rearranged as: $f_j = 1$, if $(\mathcal{M}_1 - \mathcal{M}_2) > 0$. For higher radix tests, this boundary could also be constant. However, since the upper and lower limits of the value $(\mathcal{M}_1 - \mathcal{M}_2)$ are not known for any of the individual tests, it is impossible to determine a constant boundary value that would be optimal for all motions,

actions and datasets. Therefore, a boundary value dependent on the limits of $(\mathcal{M}_1 - \mathcal{M}_2)$ are defined. Variable quantisation boundaries are employed based on $\max(\mathcal{M}_1, \mathcal{M}_2)$ and $-\max(\mathcal{M}_1, \mathcal{M}_2)$, determined on the training data.

Using higher radices with variable boundaries for the quantisation of node tests, the Relative Motion Descriptor not only captures the differences between interest point responses $\mathcal{M}_1$ and $\mathcal{M}_2$ between regions, but also the extent of the differences, as defined by the quantisation factor $\hbar$. This distinguishes between two compared regions with similar responses and regions with greater disparity in the strength of motion, and also in the negative and positive direction. The value of $\hbar$ is randomly generated, along with the positions and scales of the compared regions, and is a property of each node.

At each position of the scanning volume, the test results $\{f_1, f_2, \dots, f_{N_f}\}$ are obtained in order, and concatenated to form a binary, ternary or quaternary value of length N_f . This is in turn converted into a decimal code, δ given by,

$$\delta_{i_{XYZ}} = \sum_j^{N_f} f_j \times n^j \quad (6)$$

where n is the number of quantisation symbols and i_{XYZ} is the scanning volume instance.

4. Automatic Mode Finding For Improved Action Recognition

Instead of considering all examples of an action category label as one class, the examples are analysed to determine various inherent modes or groups. An action class can, therefore, be partitioned based on these groups, significantly simplifying the training and classification task. This is achieved by employing a variant of the RANSAC algorithm [4] to training examples of the action class.

In the example in Figure 4, it can be seen that the learning problem is highly non-linear. An attempt to learn the optimal decision boundary causes over-fitting of the non-linear classifier, and results in a complex decision boundary. For such scenarios, the outlier detection method aims to highlight and discard examples that diminish the performance of the classifier.

4.1. Outlier Detection

Given a set of training examples, Φ belonging to a particular class, C from which outliers are to be detected, a random subset, $\varphi \subset \Phi$ of the examples

(a) Illustration of overlap between positive and negative examples, resulting in a complex decision boundary. (b) Highlighting and discarding outliers simplifies the decision boundary, and reduces over fitting.

Figure 4: The overlap problem.

is iteratively selected. A classifier is then learnt of the subset φ against all training examples from other classes. This forms the hypothesis stage. The resulting model is subsequently evaluated on the remainder of the training example set, $\psi \subset \Phi$, where $\Phi = \varphi \cup \psi$ and $\varphi \cap \psi = 0$. For each iteration, $e = \{1 \dots N_e\}$, a consensus set is obtained, labelled Group ς_e , which is made up of the training subset, φ_e and the correctly classified examples in subset ψ_e .

For each iteration, the procedure identifies examples in the subset ψ that are close to examples φ in the feature space. Since the subset φ is trained against training examples from other classes, the procedure also identifies examples in class C that can be easily classified when trained against negative class examples, and those that are most likely to cause confusion with other classes and complicate class decision boundaries. After they are highlighted, the outlier subset of the training examples is discarded, and the inliers form the training set for class C . The discarding of these examples simplifies the modelling of actions by the classifier and as shown later increases classification performance.

4.2. Selecting Inliers

After several hypothesis-test iterations, the resulting consensus sets, ς are ranked by a normalised cardinality score, ϑ . This score is based on both the relevance of the set, derived from its cardinality $|\varsigma_e|$; and the frequency of occurrence of its elements within all consensus sets.

Each consensus set ς_e is given a relevance weight $\mathcal{W}_e$ obtained from the cardinality of the set, as bigger consensus sets indicate the selection of ex-

amples that are well separated from negative class examples. This consensus relevance weight is given by,

$$\mathcal{W}_e = \frac{|\varsigma_e|}{|\Phi|}, \quad (7)$$

where $|\cdot|$ denotes cardinality.

Each example Φ_i in the training set is subsequently assigned an occurrence weight ω_{Φ_i} , which is based on the number of occurrences of that example within all consensus sets, and hence its suitability as an inlier. The occurrence weight for the training example is also a function of the relevance of the consensus sets to which it belongs, and is given by,

$$\omega_{\Phi_i} = \sum_{e=1}^{N_e} \mathcal{W}_e \times \mathbf{1}_{\varsigma_e}(\Phi_i), \quad (8)$$

where $\mathbf{1}_{\varsigma_e}(\Phi_i)$ is the indicator function of training example Φ_i in consensus set ς_e , given by

$$\mathbf{1}_{\varsigma_e}(\Phi_i) = \begin{cases} 1 & \text{if } \Phi_i \in \varsigma_e; \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Based on these weights, the cardinality score ϑ_e for each consensus set is determined by,

$$\vartheta_e = \sum_{i=1}^{|\Phi|} \omega_{\Phi_i} \times \mathbf{1}_{\varsigma_e}(\Phi_i). \quad (10)$$

Given a distribution of cardinality scores for a class C , inliers $\mathcal{I}_C$ are chosen as examples occurring in the highest ranked consensus sets based on this score,

$$\mathcal{I}_C = \varsigma_1 \cup \varsigma_2 \cup \dots \cup \varsigma_{\Pi}, \quad (11)$$

where $\Pi < N_e$ is the number of ranked consensus sets chosen, and inliers constitute the training set for the class in question.

Examples of other methods which attempt to limit the influence of outliers include the soft margin extension of SVMs [31]. The soft margin method permits misclassifications of test examples via the use of slack variables for such non-separable training sets for improved class separability. The outlier detection method described here reduces the burden on such classification methods, since the method discards highlighted outliers that cause gross non-separability in the data.

4.3. Automatic Mode Finding

Outlier detection separates class examples into majority inlier and minority outlier groups. For training examples of actions with significant variations, mode finding for a particular class, C involves automatically grouping class examples into several subcategories. The process results in a number of subcategories within the class, with each one treated as a separate class during training. These subcategories highlight different modes of an action class, which, when combined, make generalisation difficult, but when analysed separately, allow for better modelling of the data.

4.4. Discovering Subcategories

In order to find the subcategories within a set of training examples, Φ of an action class, C , mean-shift clustering is first applied. This results in an initial partitioning of the data, giving non-overlapping groups $\zeta_i, i = \{1 \dots N_\zeta\}$ of the examples, as clustering does not consider separability from examples of other classes. The clustering procedure, however, provides a coarse map of groupings within the class, which serves as an initial state from which compact, more discriminative modes can be discovered.

The discovery of subcategories proceeds similarly to the outlier detection method explained in Section 4.1. However, in this case, the subset φ is selected as a random subset of one of the clusters, $\varphi \subset \zeta_i$. This ensures the compactness of the groups, and reduces the number of hypothesis-test iterations required, as ζ_i is a smaller subset of the training examples. As with outlier detection, a classifier is learnt of φ against the training examples from negative classes, $\bar{\Phi}$, giving the hypothesis phase. Testing is carried out on the remainder of training examples of class $\Phi - \varphi$, (including examples from other clusters) to produce a consensus set, ς . Numerous iterations of this process are repeated for each cluster, resulting in several consensus sets. Figures 5(a-g) illustrate the steps.

An additional benefit of clustering at the initial stage is that the discovery and selection of sub-categories degrades to the cluster groups if adequate consensus sets cannot be found.

4.5. Subcategory Selection

Given a number, Π of consensus sets generated as described above, each consensus set, $\varsigma_\pi, \pi = \{1 \dots \Pi\}$ represents a potential subcategory within the class. Since the basis of the consensus sets are groups of examples that are compact in the input space as determined by clustering, the resulting

(a) Original Classification Problem. (b) Apply mean-shift clustering.

(c) Randomly select a subset from one cluster (cluster 1, iteration 1). (d) Classify remainder of the position.

(e) Randomly select more subsets from cluster (cluster 1, iteration 2). (f) Again, classify remaining position.

(g) Process is repeated with multiple iterations for each cluster.

Figure 5: Automatic Mode Finding

consensus set can be assumed to be compact while being separable from examples of negative classes, as illustrated in Figure 5(g).

In selecting subcategories, a particular consensus set can only be selected once. AdaBoost [32] is applied in an attempt to ensure that all training examples are represented in at least one of the subcategories. To this end, each consensus set is given a score which is the sum of weights $W_\pi(i)$ associated with each example i in the group. The process is initialised by assigning equal weights, $W_1(i) = \frac{1}{|\Phi|}$ to all training examples. Hence, in the first instance, the consensus set with the highest cardinality is selected. This makes the first subcategory, labelled $\mathcal{S}_1$.

For subsequent subcategories, the weight of each example is given by

$$W_{\pi+1}(i) = \frac{W_\pi(i)}{Z_\pi} \exp(-\alpha_\pi y_i h_\pi(x_i)), \quad (12)$$

given that,

$$\alpha_\pi = \frac{1}{2} \ln\left(\frac{1 - \epsilon_\pi}{\epsilon_\pi}\right) \quad (13)$$

and the term $y_i h_\pi(x_i) = \{-1, +1\}$ denotes the absence or presence of a particular example in the previously selected subcategories, respectively. Z_π is a normalisation constant, and ϵ_π is the error rate. This selection process terminates when all examples have been chosen in at least one of the groups.

For the training of the final action classifiers, each subcategory, $\mathcal{S}_i$ is trained separately against examples of other classes. Examples of class C that do not belong to the subcategory being trained are not included in the training of that classifier. However, during classification, results of all subcategories of the class C are combined. Classification results of subcategories are combined such that a class label, $\hat{C}$ is assigned if any one of its subcategory classifiers produces the highest confidence for an example.

5. Experimental Setup

We apply the Relative Motion Descriptor to actions in video by running a scanning volume over the video. To do this efficiently, we subsampled all example videos to 120×160 pixels spatially. Since the descriptor does not make use of explicit appearance information and detected interest points are based on motion, the information lost as a result of subsampling is negligible.

We use a non-linear Support Vector Machine (SVM) in the learning of action categories. The Radial Basis Function (RBF) kernel is used, defined

by, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ represent concatenated histograms of relative motion. A multiple category classifier is constructed by combining several binary classifiers in a *one-against-rest* training scheme. Each test example is classified by all binary classifiers, returning a measure of confidence. The test example is assigned the label of the classifier that returns the highest confidence.

Training and classification for the detection of outliers and discovery of modes are carried out using SVMs. The performance of the SVM classifier is known to be limited on unbalanced data. Since our method involves training smaller positive subsets against majority negative data, we over-sample the positive class by generating synthetic training examples in feature space in order to achieve balance between the classes. We use the Synthetic Minority Over-sampling Technique of Chawla et al [33].

For Mode Finding, the initial clustering of the training examples is performed using the Mean-Shift Clustering algorithm with the Squared Euclidean distance. Yu *et al* [34] show that the kernel nearest neighbour in the RBF kernel space degenerates to conventional nearest neighbour. Therefore, since SVM classifiers with the RBF kernel are used in training, it is assumed that groups based on distance in the Euclidean space will retain their compactness in the RBF kernel space.

5.1. Datasets

Figure 6: Example actions from the Youtube action dataset.

The performance of the descriptor is evaluated on four human action datasets, namely the KTH Human Action dataset, the Assisted Daily Living data, the Kisses/Slaps dataset, and the YouTube Human Action dataset.

The KTH dataset [9] contains videos of 25 persons performing six actions in four different conditions. These actions are Boxing, Clapping, Waving, Jogging, Running and Walking. The conditions include scale variations, different clothes and lighting changes.

The Assisted Daily Living (ADL) dataset [35] consists of 150 high resolution videos of activities performed in daily living. Each action class contains 15 examples, and actions include Answer Phone, Chop Banana, Dial Phone, Look Up In Directory, Write On Whiteboard, Drink Water, Eat Snack, Peel Banana, Eat Banana, and Eat With Silverware. These actions were chosen for their similarity and the difficulty in separating between actions using only one feature type. Figure 7 shows example actions from the dataset.

Figure 7: Example actions from the Assisted Daily Living action dataset.

While both of the datasets above contain simulated actions performed in the presence of minimum background clutter, using static cameras, the Kisses/Slaps dataset [36] contains actions in two classes - Kissing and Hitting - compiled from movies. These actions are performed by different actors, at different scales, and in a wide range of scenes. Examples are shown in Figure 8. Similarly, the YouTube dataset [13] contains actions obtained from YouTube, TV broadcast, and personal video collections and are captured under uncontrolled conditions. The videos are of varying resolution, and contain significant variation. There are 11 action categories as listed in Table 6. A subset can be seen in Figure 6.

Figure 8: Selected examples of the Kisses/Slaps dataset actions.

Interest points are detected from action videos using the interest point detector of Dollar et al. [11]. Dollar applies a response function to the video of the form, $\mathfrak{R} = (I * g * h_{ev})^2 + (I * g * h_{od})^2$, where $g(x, y : \sigma)$ is the 2D Gaussian kernel applied along the spatial dimensions of the video, and h_{ev} and h_{od} are a pair of 1D Gabor filters applied in the temporal dimension. Local maxima of the response function $\mathfrak{R}$ above a threshold, are selected as interest points. This is arguably one of the simpler interest point detectors, with poorer performance than recently proposed schemes. However as we will see, the proposed classification architecture is capable of providing state of the art performance non the less.

In order to demonstrate the independence of our Mode Finding approach to the action descriptor and classification approach, an evaluation is carried out using the more recent and popular Histograms of Optical Flow (HoF) descriptor in a bag-of-words approach [12] on the Hollywood2 Human Action dataset [1]. The Hollywood2 dataset contains 12 action classes: Answer Phone, Drive Car, Eat, Fight Person, Get Out of Car, Handshake, Hug Person, Kiss, Run, Sit Down, Sit Up and Stand Up, obtained from Hollywood movies. Like the Kisses/Slap and Youtube datasets, this dataset contains significant variations in action execution and video setup across the examples, and is therefore suitable for this evaluation. The examples are split into 823 training and 884 test sequences, where training and test sequences are obtained from different movies.

For the purpose of comparison with state-of-the-art methods, the Training/Validation/Test split for the KTH as outlined in [9] is used, and the standard clean Train/Test split is used for the Hollywood2, and for the other datasets, 5 fold cross validation is used.

5.1.1. Parameter Selection

A set of parameters is determined for the Relative Motion Descriptor by optimising classification performance using the validation subset of the KTH dataset, as outlined by Schuldt *et al* [9]. Since evaluation on other datasets make use of cross validation without scope for parameter optimisation, parameters obtained from the KTH dataset are used. This ensures we are not over fitting to any of the datasets, as no optimisation is performed on any test data. For the Assisted Daily Living dataset, the same parameters are used as the dataset contains simulated actions with minimal variations in capture conditions similar to KTH. Since greater variations exist in the Kisses/Slaps and YouTube datasets, additional node test arrays, N_A are computed in order

to capture additional variation in these datasets.

The parameters optimised are Number of node tests (N_f), Number of node test arrays (N_A), Spatial and temporal size of the scanning volume ($X \times Y$ and T), and the test quantization type (binary, ternary or quaternary). Each of these parameters are chosen independently, while others are kept constant. For the scanning volume size, and type of tests, the same node tests were used in order to minimize effects that may arise as a result of randomness of the tests.

For all datasets, we use Scanning Volume Depth, $T = 35$ pixels, Scanning Volume Length $X, Y = 60$ pixels, Number of Node Tests $N_f = 4$ and Quaternary quantisation. For the KTH and Assisted Daily Living datasets, Number of Node Test Arrays $N_A = 15$ is used. For the Kisses/Slaps and YouTube datasets, we double the Number of Node Test Arrays, using $N_A = 30$, in order to capture additional variation in these datasets, selected randomly at training. The parameter learning process seeks to maximise the average classification accuracy over all classes. Where applicable, 5-fold leave-one-out cross validation is used, i.e., the training data is split into 5 subsets of equal size; and each subset is tested using a classifier learnt on the 4 remaining subsets.

Dataset	Binary	Ternary	Quaternary
Accuracy	85.6%	88.9%	93.1%
Dimension	240	1215	3840

Table 1: Comparison of the RMD vector dimension and accuracy obtained on the KTH dataset for binary, ternary and quaternary representations.

Table 1 shows the effect of different node tests on the overall performance for the KTH dataset along with the dimensionality of the vectors produced. While this might suggest higher order node tests might give higher performance, the increase in dimensionality becomes prohibitively high.

Figures 9 and reffig:Youtubeparams shows the variation of average validation accuracy for different parameters of the RMD descriptor for both the KTH and Youtube datasets respectively. Note that the characteristics of the RDM across parametrisation for both datasets are similar. With the only major difference being that the more complex Youtube data benefits from increasing the number of test arrays.

Figure 9: Variation of Average Validation Accuracy across parameters for KTH Dataset

Figure 10: Variation of Average Validation Accuracy across parameters for Youtube Dataset

Outlier Detection. For Outlier Detection, examples belonging to the highest ranked consensus sets make up the inliers for the class. The number ξ of the top consensus sets after ranking determines the tolerance of the outlier detector, where fewer consensus sets result in more outliers, and a high number of consensus sets results in a more tolerant outlier detector. This value is found to be dataset and action-specific. Hence, datasets with fewer noisy action examples generally require less tolerance, and therefore a low ξ , whereas more complex datasets require higher values. Also, within datasets, some actions are more easily distinguishable than others. For example, in the KTH dataset, static actions require a low ξ , as there is less confusion, whereas the actions *Jogging* and *Running* require more tolerance, as more confusion is observed between them.

For a large set of cardinality scores, it is assumed that the distribution is approximately Gaussian. The number of consensus sets ξ that make up the inlier subset are chosen as the number of consensus sets with cardinality scores greater than 2 standard deviations above the mean score.

Mode Finding. For Mode Finding, the most important parameter is the number of subcategories, N_S . However, this value is automatically chosen by the adapted AdaBoost reweighting scheme, which encourages the selection of all training examples, and terminates when all examples are represented in at least one group. Moreover, the non-parametric Mean-Shift clustering method is employed to determine coarse groupings ζ_i , of examples. However, within these clusters, (and within all class training examples for outlier detection) the size of the hypothesis set, $|\varphi|$ needs to be determined. The hypothesis set size is defined as a fraction of its cluster, $|\zeta_i|$ for Mode Finding, and a fraction of the training set, $|\Phi|$ for Outlier Detection.

If a higher fraction is used, the influence of each training example is diminished. Moreover, the likelihood of retaining the original decision boundaries is high. Conversely, a low fraction results in training a small percentage of the positive examples against majority negative examples. This creates significant imbalance in training, which can result in lower recall rates, despite over-sampling. However, examples that are correctly identified despite this imbalance are deemed good representative examples, and are therefore suitable as inliers or group members. Therefore, for all RANSAC hypotheses, one-third of the available examples are used. For Mode Finding, Synthetic Minority Oversampling [33] is applied to clusters prior to the RANSAC hypothesis stage, to generate additional examples in the feature space.

For Outlier Detection and Mode Finding, the number of iterations N_e , needs to be chosen such that an over-complete set of random combinations of the RANSAC hypothesis and test examples are chosen. This number therefore depends on the size of the training set for the class. In order to obtain a large variety of combinations, $N_e = 100 \times |\Phi|$ hypothesis-test iterations are conducted. For the datasets on which the method is evaluated, the maximum class training sample size is 195 for the action *Horse Riding* in the YouTube dataset, resulting in 19500 iterations for that class.

Mode Finding with a HOF Bag-of-words approach. For this experiment, the experimental setup of Laptev et al. [12] is followed to obtain and evaluate Histograms of Optical Flow (HoF): Interest points are detected using the spatio-temporal extension of the Harris detector, and descriptors of the spatio-temporal neighbourhoods of the interest points are computed, using the parameters proposed by Laptev. Clusters of 4000 visual words are obtained from a subset of 100,000 interest points, using k-means with the Euclidean distance. Non-linear Support Vector Machine classifiers with a χ^2 kernel are learnt using a binary one-vs-rest training scheme, and performance of the method is evaluated by computing the Mean Average Precision over the binary problems.

Having trained using the more compact sub-categories, during testing, we obtain confidence scores from all sub-category binary classifiers for each test example. In order to obtain average precision values which combine results of multiple sub-categories within a class, we normalise the scores, such that the values are distributed over a range of $[0, 1]$, and make use of a single threshold across the multiple sub-category scores within that range. Precision-Recall curves which combine the results of the sub-categories are generated by varying this single threshold, and using the logical-OR operator across sub-categories, on the label given to each test example. In particular, for each increment of the threshold, positives, from which precision and recall values are obtained, are counted for the class if *any* one of its sub-category scores is above the threshold.

6. Results

6.1. KTH Human Action Dataset

For the KTH Human Action dataset, using the training/validation/test split defined by [9], an overall accuracy of 91.2% was obtained with the

introduced Relative Motion Descriptor. Figure 11 show the confusion matrix of the classification. It can be seen that static actions are well separated from dynamic actions. That is, while some confusion is observed between Boxing, Handclapping and Handwaving, and between Jogging, Running and Walking, no confusion is observed between the two groups of actions. This shows the descriptor’s ability to encode different types of motion. Most confusion is observed between the action classes, Jogging and Running. This is attributed to similarities in the execution of both action classes, and thus, the underlying motions between the two classes. This confusion is typically seen in competing state of the art approaches.

Figure 11: Confusion Matrix for the Relative Motion Descriptor on the KTH dataset. Average classification accuracy is 91.2%.

Table 2 compares average accuracy obtained using the Relative Motion Descriptor with state-of-the-art methods using the same Training/Validation/Test methodology. Given that the interest point detector of Dollar *et al* [11] was used, the Relative Motion Descriptor achieves an average accuracy 10% greater than Dollar’s approach, which employs concatenated vectors of gradients of interest points in conjunction with SVM classifiers. Also, the results show comparable performance to descriptors which make use of both ap-

pearance and motion information, and highlight the discriminative power of the relative motion representation. When compared with other motion oriented approaches, it is observed that the encoding of relative motion at local regions across action sequences provide better discriminatory information.

Method	Accuracy
Schuldt <i>et al.</i> [9]	71.72%
Dollar <i>et al.</i> [11]	81.2%
Reddy, Shah [22]	89.79%
Fathi and Mori [37]	90.5%
Klaser <i>et al.</i> [18]	91.4%
Marszalek <i>et al.</i> [12]	91.8%
egonzio <i>et al.</i> [19]	94.33%
Kovashka <i>et al.</i> [27]	94.5%
Gilbert <i>et al.</i> [14]	94.5%
RMD	91.2%
RMD + Outlier Detection	94%
RMD + Mode Finding	92.1%

Table 2: Average accuracies on the KTH dataset using the Training/Validation/Test split defined in [9].

Figure 12(a) shows the confusion matrix obtained using Outlier Detection on the KTH dataset, and the effects of discarding noisy examples from training can be observed. For the static actions, the elimination of outliers has no effect on the classification accuracy. However, significant improvements are made on the classification accuracy of the three dynamic actions with the Running and Walking classes benefiting the most from the method. The separation of the Running and Jogging classes has historically proven difficult in this dataset, as a result of the similarities in the execution of the actions. These results show that the removal of examples from training that appear too similar to other class examples, improves the performance of both classifiers. Similarly, Outlier Detection reduces the misclassification of Walking examples as Jogging. The overall accuracy on the KTH dataset using the Relative Motion Descriptor with Outlier Detection is 94%, which is an increase of 3% over the standard Relative Motion Descriptor.

Figure 12(b) shows results obtained by splitting training examples into sub-categories of their original class labels. While there is an overall im-

Figure 12: Confusion Matrices for the RMD Descriptor on the KTH dataset after Outlier Detection and Mode Finding.

provement in results over the Relative Motion Descriptor, at 92.1%, the gains obtained are not as significant as for Outlier Detection. The limited gains obtained with grouping are not unexpected as the KTH dataset is limited in variability. Hence, the splitting of certain class example sets serve to weaken some of the classifiers since the dataset is not multi-modal, given the motion descriptor used.

Table 2 compares average accuracy obtained using the Relative Motion Descriptor, Outlier Detection and Mode Finding with state-of-the-art methods using the same Training/Validation/Test methodology of Schuld *et al* [9]. Table 3 shows the number of consensus sets, $|\xi|$ for Outlier Detection, the number of outliers discarded, $|\Theta|$, and for Mode Finding, the number of subcategories N_S , detected for each class in the KTH dataset.

Having performed the Outlier Detector procedure on all classes of the dataset, it can be seen that no outliers were obtained for the actions *Handwaving* and *Walking*, while *Boxing* discarded 2 training examples, and 1 example was discarded from *Running* and *Handclapping*. For the *Jogging* action, 6 outliers were discarded. Some were discarded as a result of interest points detected on the shadow of subject, while others were as a result of

Action	Number of sets, $ \xi $	Number of Outliers, $ \Theta $	Number of Subcategories, N_S
Boxing	3	2	2
Handclapping	3	1	2
Handwaving	3	0	1
Jogging	5	6	3
Running	4	1	2
Walking	2	0	1

Table 3: Optimal numbers of consensus sets $|\xi|$, and number of outliers $|\Theta|$ for Outlier Detection; and subcategories N_S , discovered for Mode Finding, for all classes of the KTH dataset.

strong interest point detections on clothing.

The results show, as for Boxing and Handclapping, that in some cases the removal of a small number of examples do not affect the class confusion. For Walking, the removal of Jogging outliers results in improved performance. However, while reduced in effect, inherent confusion still exists between the Jogging and Running classes. This is due to the actions being indistinguishable in many of the examples, and despite the simplification of the decision boundary between these classes, misclassifications still occur.

Figure 13(b) shows examples of outliers discarded by the RANSAC procedure. It is observed that some outliers are regarded as such because of interest points detected on shadows of the subject, which affect the distribution. For translational actions of the dataset, videos with scale change constitute some of the outliers, while the motion for other examples are slower compared to the majority of examples. For example, where the Running action is performed at a slower speed, it is confused with Jogging. These two classes are shown to produce the most outliers. Figure 13(c) shows examples from the two discovered groups of the jogging action. The second group consists of scale change examples, where the motion is diagonal across the field of view, and thus, appears slower in the motion descriptor.

6.2. Assisted Daily Living Dataset

For the Assisted Daily Living dataset, 5-fold cross validation is used (4 folds for training 1 for testing), and an overall recognition accuracy of 89.3% was obtained. Figure 14 shows the confusion matrix for the dataset. Given

(a) Examples of outliers selected from the Jogging action of the KTH dataset. A majority of the outliers are caused by the existence of moving shadows in the examples, causing variations in the motion descriptor, while others are as a result of variations caused by interest points induced on clothing.

(b) Examples of outliers selected from other classes. Left: Running action with interest points detected on shadow; Middle: Boxing action with interest points detected on shadow; Right: A Handclapping action performed differently from other examples.

(c) Non-overlapping examples within two groups of the Jogging action from the KTH dataset. Left: Normal scenario of the action; Right: Scale change scenario, where the subject moves diagonally across the screen

Figure 13: Detected outliers and modes from the KTH dataset.

that the majority of the actions in this dataset involve very similar motions, and were chosen for the difficulty in categorising them using a single feature type [35] (*Drink Water, Eat Banana, Eat snack, Use Silverware, Answer Phone*), the results obtained are impressive.

Figure 14: Confusion Matrices for the Relative Motion Descriptor on the Assisted Daily Living dataset. Average classification accuracy is 89.3%.

As no explicit appearance information is used, the success of the Relative Motion Descriptor in categorising these actions is attributed to the use of response strengths of detected interest points and the additional quantisation of node test results based on comparisons of the intensities of motion. Hence, while the motion may be similar, subtle differences in the nature of the motion, which are encoded by multiple quantisation symbols provide additional discriminatory cues for categorisation. It should be noted that the Relative Motion Descriptor outperforms other single-feature approaches on this dataset. Table 4 shows a comparison with other methods on this dataset.

The use of Outlier Detection in conjunction with the Relative Motion Descriptor improves the average accuracy to 92%, while Mode Finding returns an accuracy of 90.7%. Figure 15(a) shows that the removal of outliers from training improves 5 of the 10 classes, while marginally degrading the performance of two classifiers. For this dataset, the outlier elimination process resulted in one outlier from the classes *Dial Phone*, *Peel Banana* and *Drink*

Water. This provides an improvement in classification where confusion originally existed with these actions. An example of this are the actions *Answer Phone* and *Use Silverware*, whose classification accuracies are improved from 86.7% to 100%. However, the removal of these outliers also results in greater confusion with the actions *Eat Banana* and *Write On Board*, though to a lesser extent. The result of this is an overall improvement in classification performance.

Figure 15: Confusion Matrices for the RMD Descriptor on the Assisted Daily Living dataset after Outlier Detection and Mode Finding.

Similarly to the KTH dataset, Outlier Detection achieves better performance than Mode Finding. This is again attributed to the limited variability in the dataset. Of the 10 action classes, 6 retain their single groupings, while the actions *Peel Banana*, *Use Silverware*, *Drink Water* and *Dial Phone* are each split in two groups. These actions involve more variable motions in the dataset. Figure 15(b) shows the confusion matrix for Mode Finding on the dataset. Improvements are observed on the actions *Answer Phone*, *Eat Snack*, *Peel Banana* and *Write On Board*, at the expense of the reduction in the accuracies of *Chop Banana* and *Drink Water*. For *Dial Phone*, the splitting of the examples results in a stronger classifier, as it can be observed that more classes exhibit confusion with it. However, in some cases, the splitting

of class examples also serves to strengthen other classifiers, as with *Drink Water*. Table 4 shows a comparison with other methods on this dataset, with all variants of the approach outperforming the state of the art.

Method	Accuracy
Velocity Histories [35]	63%
Latent Velocity Histories [35]	63%
Tracklets [38]	82.7%
Augmented Velocity Histories with Relative and Absolute Position [35]	89%
Relative Motion Descriptor	89.3%
RMD + Outlier Detection	92%
RMD + Mode Finding	90.7%

Table 4: Comparison on Assisted Daily Living dataset.

6.3. Kisses/Slaps Dataset

The KTH and Assisted Daily Living datasets provide a good basis to test the Relative Motion Descriptor. However, they exhibit minimal variation in camera setup and action execution. Additional results are therefore presented on the Kisses/Slaps, YouTube and Hollywood2 human action datasets. As mentioned, actions in these datasets are performed in natural settings, and no predefined constraints exist for the execution of actions or camera setup. For these datasets, consistent descriptors of relative motion across examples are much more difficult to obtain as a result of the significant amounts of variation present. However, the use of additional arrays of node tests encodes additional motion information.

For the Kisses/Slaps dataset, an overall accuracy of 77.6% is obtained using 5-fold cross validation. This is a considerable improvement on the Action MACH approach of Rodriguez *et al.* [36]. These results show that the descriptor performs well at encoding actions in complex settings. Also, despite the presence of interest point detections due to camera motion and cluttered backgrounds, the representation is capable of identifying characteristic motions in such settings.

Table 5 shows results obtained on the binary Kisses/Slaps dataset with Outlier Detection and Mode Finding. For Outlier Detection, the overall recognition accuracy is improved from 77.6% to 86.9%, with class accuracies

of 89.5% and 84.3% for the Kisses and Slaps actions respectively. This gives an increase of 12% over both classes. Using 5-fold cross validation, the number of outliers for all leave-out iterations are 9 and 4 for Kisses and Slaps respectively. By automatically splitting the training examples into their in-

Method	Actions(%)		Average
	Kisses	Slaps	
Action MACH [36]	66.4%	67.2%	66.8%
Local Trinary Patterns [39]	77.3%	84.2%	80.75%
Relative Motion Descriptor	76.1%	79.1%	77.6%
RMD + Outlier Detection	89.5%	84.3%	86.9%
RMD + Mode Finding	90.4%	88.7%	89.1%

Table 5: Comparison on the Kisses/Slaps dataset.

herent groups, the overall accuracy is further improved to 89.1%. Both classes return a remarkable increase over the Relative Motion Descriptor, and show that, in contrast to simulated actions in simplified settings, the use of grouping hugely benefits actions with larger amounts of variation. Table 5 also compares the results to other methods. For Mode Finding, 6 and 4 groups are discovered for Kisses and Slaps respectively. Figure 16 show examples in two groups of the Slaps class. In one of the subcategories (Figure 16(a)), the *Hitting* action occurs with minimal background motion, whereas in the subcategory with examples shown in Figure 16(b), the backgrounds are seen to be more active.

6.4. YouTube Action Dataset

The Kisses/Slaps dataset is limited in the number of classes. The YouTube dataset contains 11 classes, and gives a better indication of performance over a greater range of classes. Figure 17 shows a confusion matrix for actions in the YouTube Action dataset. As expected, and as observed with previous datasets, greater confusion is observed between actions that involve similar motion. The greatest confusion is obtained between *Volleyball* and *Basketball*, which both involve tossing a ball in the air and the movement of several players across the camera view. Confusion is also observed between actions involving translational motion, and often camera motion, for example, *Walk Dog*, *Bike Riding*, *Horse Riding*; and also between *Golf Swing* and *Tennis*, which involve a swinging motion.

(a) Examples belonging to one subcategory of the Slaps class of the Kisses/Slaps dataset.

(b) Examples belonging to another subcategory of the Slaps class of the Kisses/Slaps dataset.

Figure 16: Examples of non-overlapping subcategories within the Kisses/Slaps dataset.

Table 6 shows percentage accuracies obtained for each class, and compares with other recent action recognition approaches on this dataset. Highlighted in the table is the number of feature types used by each approach. The single-feature Relative Motion Descriptor obtains an accuracy of 71.9%. Compared with other single feature approaches, such as Liu et al 's [13] individual static and motion features obtain accuracies of 63.1% and 65.4% respectively. The single feature Relative Motion Descriptor (RMD) outperforms Liu's single comparable motion feature by 6%. Recently Ikizler-Cinbis [40] showed good performance however, they performed computationally intense steps such as video stabilization, persons detection, and tracking. Similarly the approach by Reddy and Shah [22] shows good performance of the YouTube dataset. However, this is in part due to the introduction of a scene context descriptor in addition to motion descriptors, as by only using the three motion descriptors, their performance drops to 59.89%.

Table 6 shows classification results on the YouTube dataset using Outlier Detection and Mode Finding with the Relative Motion Descriptor. The benefits of Outlier Detection can be observed on 6 of the 11 classes, as improvements are made over the Relative Motion Descriptor results and other approaches for these classes. Figure 18(a) shows the confusion. In the ma-

WalkDog	17.9	1.6	4.1	5.7	1.6	14.6	0.8	2.4	4.1	0.8	46.3
Tennis		4.2	5.4	5.4	1.8	1.2	4.2	1.8	1.2	73.7	1.2
Swing	4.4	5.1	0.7	1.5	2.9			3.6	76.6	0.7	4.4
VBall	0.9	11.2	2.6	3.4	3.4	2.6	17.2	50.0	3.4	5.2	
BBball	0.7	5.7	5.0	2.1	0.7	4.3	54.6	11.3	2.8	12.1	0.7
Horse	6.2	1.0			2.6	82.1	1.5		1.0	1.5	4.1
Tramp	3.4			1.7	82.4	0.8		2.5	3.4	5.0	0.8
Juggle		1.3		76.9	3.8	1.9	2.6	1.3	3.2	6.4	2.6
Golf		2.8	82.4	2.1			1.4	0.7		9.9	0.7
Dive	1.9	79.5	1.9	0.6	0.6	3.2	3.8	2.6	1.3	2.6	1.9
Cycle	75.2	0.7		0.7	0.7	9.0	1.4	0.7	2.8	1.4	7.6
	Cycle	Dive	Golf	Ball Juggle	Trampoline	Horse Ride	Basketball	Volleyball	Swing	Tennis	Walk Dog

Figure 17: Confusion Matrix for the YouTube Human Action dataset. Confusion is displayed across. Overall Accuracy=71.9%.

	Percent accuracy (%)						
Action	[13]	[40]	[22]	[22]	RMD	RMD + Out. Det.	RMD + Modes
Cycling	73	75.2	49	-	75.2	71.5	84.1
Diving	81	95	81	-	79.5	80.8	87.1
Golf	86	95	80	-	82.4	84.5	79.3
Ball Juggle	54	53	56	-	76.9	73.7	82.5
Trampolining	79	93	61	-	82.4	76.5	87.3
Horse Riding	72	73	55	-	82.1	83.6	92.3
Basketball	53	48.5	56	-	54.6	52.5	65.5
Volleyball	73.3	85	71	-	50	50.9	71.3
Swing	57	66	55	-	76.6	80.3	87.5
Tennis	80	77	59	-	73.7	77.8	82.5
Walk Dog	75	66.7	36	-	46.3	52.5	72.1
No. of Feats	2	3	3	3+Scene	1	1	1
Average (%)	71.2	75.2	59.89%	85.34	71.9	72.4	81.7

Table 6: Percentage Accuracy on the YouTube action dataset.

jority of cases, for example *Walk Dog*, *Basketball Shooting* and *Horse Riding*, the increase in accuracy corresponds to a reduction in confusion with the most similar negative classes, as previously shown by the RMD confusion matrix in Figure 17 and as observed for the KTH dataset. These results show that improvement from the simplification of class boundaries extends to actions performed in complex settings. Reduction in accuracy is observed for some classes, for example *Cycling*. This can be attributed to the creation of stronger classifiers of negative classes in the absence of noisy examples. Also, the redefinition of class boundaries will, in some cases, favour one action over another, when outliers are discarded. For such cases, Mode Finding seeks multiple boundaries.

Using Mode Finding, an increase in accuracy of 9.8% is observed. Improvements in accuracy are obtained on all but one of the classes over the Relative Motion Descriptor, and further reduction in confusion is observed between actions with similar motions, as shown in Figure 18(b). For example, *Volleyball* and *Basketball* show an increase of 21% and 11% respectively, while the *Walk Dog*, *Bike Riding* and *Horse Riding* actions also benefit from

significant classification accuracy increases.

Figure 18: Confusion Matrices for the RMD Descriptor on the Youtube dataset after Outlier Detection and Mode Finding.

Figure 19 show examples of visually discernible groups discovered automatically for the *Cycling* action. Grouped in the depicted subcategories are examples with a static handled camera with minimal jitter, examples of a moving camera alongside cyclist, and examples of a moving camera behind cyclist. The groups show different distributions captured by the Relative Motion Descriptor, and the ability of the Mode Finding method to discover the various motion patterns.

6.5. Hollywood2 Action Dataset

Table 7 shows the average precision obtained for each class using the Histograms of Optical Flow descriptor [1], along with results obtained when used in conjunction with the Outlier Detection and Mode Finding approaches. Figure 20 show precision-recall curves for all classes of the HoF descriptor.

As with the Kisses/Slaps and YouTube datasets, it can be seen that the removal of outliers improves classification performance for the majority of classes: 9 out of 12 classes indicate improvements in average precision. For these classes, the benefit of simpler classifier boundaries is evident, as the confidences returned for the class test examples are improved. An increase

Action	Num. train.	Average Precision		
		HoF [1]	HoF+ Outl.	HoF+ Modes
AnswerPhone	66	0.087	0.098	0.182
DriveCar	85	0.825	0.829	0.856
Eat	40	0.566	0.517	0.562
FightPerson	54	0.625	0.629	0.571
GetOutCar	51	0.086	0.120	0.142
HandShake	32	0.172	0.123	0.140
HugPerson	64	0.198	0.220	0.214
Kiss	114	0.320	0.340	0.396
Run	135	0.662	0.653	0.684
SitDown	104	0.413	0.420	0.532
SitUp	24	0.068	0.092	0.127
StandUp	132	0.401	0.405	0.520
Mean		0.368	0.371	0.410

Table 7: Average Precision for all classes on the Hollywood2 dataset, using HoF, HoF with Outlier Detection, and HoF with Mode Finding.

Figure 19: Examples of non-overlapping subcategories within the *Cycling* class of the YouTube dataset. *Left*: Static camera with minimal jitter; *Middle*: Camera moving alongside cyclist; *Right*: Camera moving behind cyclist.

in Mean Average Precision is observed, from 0.368 for the Histograms of Optical flow descriptor, to 0.371 with the introduction of Outlier Detection. The action *GetOutCar* and *SitUp* benefit the most from Outlier Detection, with percentage increases of 39.5% and 35% respectively. For the classes *Eat*, *HandShake* and *Run*, reduction in average precision are observed as a result of the removal of outliers from training. This is attributed to the removal of examples that contributed significantly to the original classifiers.

A mean average precision of 0.41 is obtained with the discovery of modes within the classes. This further demonstrates the benefits of assuming multi-modality in complex action classes. Moreover, this shows that the Outlier Detection and Mode Finding approaches are applicable to any feature type and encoding. In addition to observations made on previous datasets, the best Mode Finding performance on this dataset is observed with classes that exhibited the worst performance using the original class groupings, where further improvements are made over Outlier Detection. An improvement of 109% is observed on the *AnswerPhone* action, while *SitUp* and *GetOutCar* show increases in average precision of 86.8% and 65.1% respectively. Other notable improvements include *Kiss*, *SitDown* and *StandUp*. These improve-

Figure 20: Precision-Recall curves for all classes of the Hollywood2 dataset, using the HoF Descriptor.

ments are due to discovery of multiple modes in training. Figure 22 shows much-improved precision-recall curves for these classes. Reduction in performance can be observed for the actions *Eat*, *FightPerson* and *HandShake*.

Figure 21 shows visually apparent and non-overlapping examples within two groups of the *GetOutCar* action. Figure 21(a) shows examples within a group where the action is captured in low illumination, and most of the examples are viewed from the inside of the vehicle with a dominant upwards motion, while Figure 21(a) depict examples where the motion of the subject is towards the left of the scene.

(a) Examples belonging to one subcategory of the *GetOutCar* class of the Hollywood2 dataset.

(b) Examples belonging to another subcategory of the *GetOutCar* class of the Hollywood2 dataset.

Figure 21: Examples of non-overlapping subcategories within the Hollywood2 dataset.

7. Conclusion

This paper presented a novel descriptor for actions in video, which solely encodes the dynamics of body movements using the relative distribution of motion-induced interest points. This relative distribution is captured using arrays of efficient node tests from which concatenated relative motion histograms are assembled. When used in conjunction with a non-linear discriminative classifier, results indicate remarkable performance compared to recent approaches which combine multiple features. Given that no explicit appearance information is used, we show that discriminative action descriptors can be created based on the dynamics of body movement alone.

This approach also highlights the benefit of quantizing the relative motion representation with more symbols using a quaternary representation of node

Figure 22: Precision-Recall curves for HOF, HOF with Outlier Detection and HOF with Mode Finding, for all classes of the Hollywood2 Dataset.

tests, thereby capturing additional information. As demonstrated by the results of the Assisted Daily Living dataset, the Relative Motion Descriptor is capable of distinguishing between actions involving similar motion, as further discriminatory information present in the response strengths of motion-induced interest points is captured in the node tests, and the results are quantised to maximise the information obtained.

We also show consistent improvements in performance over all evaluated datasets by discarding training examples that complicate the decision boundaries between classes. These examples are discovered by our novel application of RANSAC in this domain. For actions in the wild, we argue that treating all examples of a semantic action category as one class is not optimal, and show that gains in performance can be achieved by identifying various modes of action execution or camera set-up. Our results show remarkable improvements in classification performance and we show that these performance gains are independent of the features used or the encoding and classification architecture. This is attributed to the learning of multiple classifiers on smaller, better-defined sub-categories within each of the classes. Our approach is generic, and can be used in conjunction with existing action recognition methods, and on complex datasets.

8. Acknowledgements

This work is supported by the EPSRC projects *Making Sense* (EP/H023135/1) and *Learning to Recognise Dynamic Visual Content from Broadcast Footage* (EP/I011811/1).

- [1] M. Marszalek, I. Laptev, C. Schmid, Actions in context, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2929–2936.
- [2] J. Liu, M. Shah, Learning human actions via information maximization, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
- [3] G. Johansson, Vision motion perception, *Scientific American* 232 (6) (1975) 76–88.
- [4] M. A. Fischler, R. C. Bolles, Ransac: A paradigm for model fitting with applications to image analysis and automated cartography, *Comms. of the ACM* 24 (6) (1981) 381–395.

- [5] H. Zhang, A. Berg, M. Maire, J. Malik, Svm-knn: Discriminative nearest neighbor classification for visual category recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2126–2136.
- [6] T. Malisiewicz, A. Gupta, A. A. Efros, Ensemble of exemplar-svms for object detection and beyond, in: ICCV, 2011.
- [7] R. Urtasun, T. Darrell, Sparse probabilistic regression for activity-independent human pose inference, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
- [8] G. Szepannek, J. Schiffner, J. Wilson, C. Weihs, Local modelling in classification, in: Proceedings of IEEE International Conference on Data Mining, 2008, pp. 153–164.
- [9] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: A local SVM approach, in: Proceedings of IAPR International Conference on Pattern Recognition, 2004, pp. 32–36.
- [10] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, *International Journal of Computer Vision* 79 (3) (2008) 299–318.
- [11] G. C. P. Dollár, V. Rabaud, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Proceedings IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72.
- [12] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
- [13] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1–8.
- [14] A. Gilbert, J. Illingworth, R. Bowden, Action recognition using mined hierarchical compound features, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99 (PrePrints). doi:<http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.144>.

- [15] I. Laptev, T. Lindeberg, Space-time interest points, in: Proceedings of IEEE International Conference on Computer Vision, 2003, pp. 432–439.
- [16] G. Willems, T. Tuytelaars, L. J. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: Proceedings of European Conference on Computer Vision, 2008, pp. 650–663.
- [17] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of International Conference on Multimedia, 2007, pp. 357–360.
- [18] A. Kläser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: Proceedings of British Machine Vision Conference, 2008, pp. 995–1004.
- [19] M. Bregonzio, T. Xiang, S. Gong, Fusing appearance and distribution information of interest points for action recognition, *Pattern Recognition* 45 (3) (2012) 1220–1234.
- [20] I. Laptev, P. Perez, Retrieving actions in movies, in: Proceedings of IEEE International Conference on Computer Vision, 2007, pp. 1–8.
- [21] D. Han, L. Bo, C. Sminchisescu, Selection and context for action recognition, in: Proceedings of IEEE International Conference on Computer Vision, 2009.
- [22] K. K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, *Machine Vision and Applications* (2013) 1–11.
- [23] M. M. Ullah, S. Parizi, I. Laptev, Improving bag-of-features action recognition with non-local cues, in: Proceedings of British Machine Vision Conference, 2010, pp. 1–11.
- [24] O. Kliper-Gross, Y. Gurovich, T. Hassner, L. Wolf, Motion interchange patterns for action recognition in unconstrained videos, in: *Computer Vision–ECCV 2012*, 2012, pp. 256–269.
- [25] M. Ryoo, J. Aggarwal, Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities, in: Proceedings of IEEE International Conference on Computer Vision, 2009, pp. 1593–1600.

- [26] P. Matikainen, M. Herbert, R. Sukthankar, Representing pairwise spatial and temporal relations for action recognition, in: Proceedings of European Conference on Computer Vision, 2010.
- [27] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010.
- [28] S. Savarese, A. DelPozo, J. Niebles, L. Fei-Fei, Spatial-temporal correlations for unsupervised action classification, in: Proceedings of IEEE Workshop on Motion and Video Computing, 2008, pp. 1–8.
- [29] T. Guha, R. K. Ward, Learning sparse representations for human action recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (8) (2012) 1576–1588.
- [30] M. Ozuysal, P. Fua, V. Lepetit, Fast keypoint recognition in ten lines of code, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
- [31] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
- [32] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of European Conference on Computational Learning Theory, 1995, pp. 23–37.
- [33] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, Smote: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.
- [34] K. Yu, L. Ji, X. Zhang, Kernel nearest-neighbor algorithm, Neural Processing Letters 15 (2002) 147–156.
- [35] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in: Proceedings of IEEE International Conference on Computer Vision, 2009.
- [36] M. Rodriguez, J. Ahmed, M. Shah, Action mach: A spatio-temporal maximum average correlation height filter for action recognition, in:

Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

- [37] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
- [38] M. Raptis, S. Soatto, Tracklet descriptors for action modeling and video analysis, in: Proceedings of European Conference on Computer Vision, 2010.
- [39] I. Yeffet, L. Wolf, Local trinary patterns for human action recognition, in: Proceedings of IEEE International Conference on Computer Vision, 2009, pp. 492–497.
- [40] N. Ikizler-Cinbis, S. Sclaroff, Object, scene and actions: Combining multiple features for human action recognition, in: Proceedings of European Conference on Computer Vision, 2010.