Action Recognition using Mined Hierarchical Compound Features

Andrew Gilbert *Member, IEEE,* John Illingworth, Richard Bowden *Senior Member, IEEE,*

Abstract—The field of Action Recognition has seen a large increase in activity in recent years. Much of the progress has been through incorporating ideas from single frame object recognition and adapting them for temporal based action recognition. Inspired by the success of interest points in the 2D spatial domain, their 3D (space-time) counterparts typically form the basic components used to describe actions, and in action recognition, the features used are often engineered to fire sparsely. This is to ensure the problem is tractable; however, this can sacrifice recognition accuracy as it cannot be assumed that the optimum features in terms of class discrimination are obtained from this approach. In contrast, we propose to initially use an over complete set of simple 2D corners in both space and time. These are grouped spatially and temporally using a hierarchical process, with an increasing search area. At each stage of the hierarchy, the most distinctive and descriptive features are learnt efficiently through data mining. This allows large amounts of data to be searched for frequently reoccurring patterns of features. At each level of the hierarchy, the mined compound features become more complex, discriminative and sparse. This results in fast, accurate recognition with real-time performance on high resolution video. As the compound features are constructed and selected based upon their ability to discriminate, their speed and accuracy increase at each level of the hierarchy. The approach is tested on four state-of-the-art datasets, the popular *KTH* dataset to provide a comparison with other state-of-the-art approaches, the *Multi-KTH* dataset to illustrate performance at simultaneous multi-action classification, despite no explicit localization information provided during training. Finally, the recent *Hollywood* and *Hollywood2* datasets, provide challenging complex actions taken from commercial movie sequences. For all four datasets, the proposed hierarchical approach out performs all other methods reported thus far in the literature, and can achieve real-time operation.

Index Terms—Action Recognition, Data Mining, Real-time, Learning, Spatio-temporal

✦

1 INTRODUCTION

THE quantity of video data containing human action is constantly growing, not only in terms of TV and movie footage but also with the revolution in personal video recording for upload to sites such as *YouTube^{TM}* or *Google videos^{TM}*. With this growth comes the need for automatic video analysis and the recognition of events. Often, major events are delineated by actions, for example, the scoring of a goal, two people hugging, or some furtive behaviour in a surveillance image, examples of which are shown in Figure 1. Many approaches to
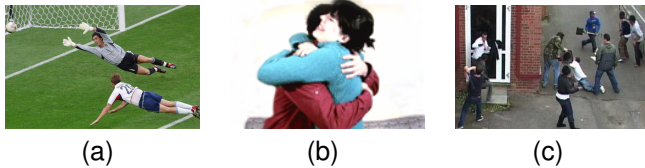

Fig. 1. Examples of human actions; (a) scoring of a football goal, (b) people hugging, (c) a fight scene

the recognition of actions extend object recognition approaches. The two problems have many shared aspects, including the necessity to handle significant within-class variation, occlusions, viewpoint, illumination and scale

- *A. Gilbert, J. Illingworth and R.Bowden are with the Centre for Vision Speech and Signal Processing, Uni. of Surrey, Guildford, GU2 7XH, UK. E-mail: {a.gilbert, j.illingworth, r.bowden}@surrey.ac.uk*

changes as well as the presence of background clutter. Figure 2(a-b) illustrates some of these issues with two

Fig. 2. Examples of *Running* from the *KTH* dataset [1]

radically different images of people running. Similarly, Figure 17 shows the variability of the action *Sit Up* from the *Hollywood* dataset [2]. In the context of object recognition, it is popular to represent an object as a *bag* of visual *words* via a histogram [3]. This histogram of words can then be used in a classifier architecture to discriminate against other classes of objects. However, not all words will be informative in terms of describing the object's within-class variation while discriminating against between-class variation. This makes the selection of the most informative words vital. The common methods of selecting words are through machine learning techniques such as Boosting [4] or Support Vector Machines [1] and adaptations such as Multiple Instance Learning (MIL)[5]. While these approaches can provide excellent results for object recognition, it has not been shown that they can be directly transferred into the temporal domain, for action recognition, without compromise. In order to scale to the

temporal domain, features are typically engineered to occur sparsely to reduce computational overheads [6]. This allows the representation to be tractable and features are assumed to be the most descriptive for the learnt actions. However, sparse features may disregard important between-class discriminatory information causing a reduction in performance. Recent studies in both the spatial [7] and temporal [8] domains explore the variability in the descriptive/discriminative power of such features.

An alternative approach is to use a more exhaustive feature set and dense features have proved beneficial[9]. However, this places considerable computational demands upon the feature selection process and therefore different methods are required to learn the action representations. Data mining provides such a method and can process vast quantities of data in an efficient and effective manner. Data mining has been successfully used in recent work [10], [11], [12], [13]. Specifically, we propose the use of data mining to allow a multi stage classifier to be learnt from a large over-complete set of simple features. The simple features are grouped both spatially and temporally into compound features of increasing complexity. Initially, a very localised neighbourhood grouping is used but then the volume of the neighbourhood is increased at each level. To allow for scale invariance, only the relative position and scale of compound features are used. By using a hierarchical approach, complex actions such as *kissing* or *getting out of a car* can be modelled in near real-time. In addition, due to the exhaustive nature of the search, features that maximise class discrimination are found. In experiments, we show that the method out performs other state-of-the-art approaches on a range of popular human action recognition datasets, including the *KTH* dataset [1] and the more challenging *Hollywood* [2] and *Hollywood2* [14] datasets. In addition, the *Multi-KTH* [15] dataset is used to demonstrate performance at classifying and localising multiple actions in noisy cluttered scenes containing camera motion.

In this manuscript we build upon our previous work in [11] and [13]. We generalise the use of the hierarchy and provide a more detailed formalisation of the stages of the approach. Within the results section there is extensive analysis and validation of the approach on four increasingly complex datasets, we also provide insight into the effect of encoding strategy, hierarchy level and feature complexity on speed and accuracy. The paper is organised as follows. Initially, an overview of recent related work is given in section 2, while section 3 explains the basic approach. Data mining is presented in section 4 and the detection and hierarchical grouping of features is explained in sections 5 and 6. Extensive results and conclusions are presented in sections 7 and 8 respectively.

2 RELATED WORK

Within the field of object recognition, the use of the spatial representation of local features has shown considerable success [10], [16], [17] and has been extended to the temporal recognition of actions. However, due to data constraints, the methods typically use a sparse selection of local interest points. Scovanner *et al* [6] extended the 2D SIFT descriptor [18] into three dimensions, by adding a further dimension to the orientation histogram. This encodes temporal information enabling it to outperform the 2D version in action recognition. Similarly, Willems *et al* [19] extended the SURF descriptor to the spatio-temporal domain. Schüldt *et al* [1] and Dollar *et al* [20] employ sparse spatio-temporal features for the recognition of human (and mice) actions. Schüldt takes the codebook and bag-of-words approach, often applied to object recognition, to produce a histogram of informative words for each action. Similarly, Dollar takes the bag of words approach but argues for an even sparser sampling of the interest points. Niebles and Fei-Fei [21] introduce hierarchical modelling that can be characterised as a constellation of bags-of-words. The hierarchical modelling provides improved performance.

Much of the early work in action recognition was tested on relatively simple, single person, uniform background sequences [1], [22]. However, these datasets are simplistic and therefore unrealistic. To address this deficiency, more natural and diverse video datasets are currently being developed. Laptev *et al* [23] expanded the ideas proposed by Ke *et al* [24] to apply volumetric features to optical flow [25], [26]. Uemura *et al* [15] used a motion model based on optical flow combined with SIFT feature correlation in order to accurately classify multiple actions on a sequence containing large motion and scale changes. Laptev [23] both exploits the motion (Histogram of optical flow, (HoF)) and appearance (Histogram of Orientation, (HoG)) of the actions, creating a boosted action classifier for recognising the human actions of smoking and drinking. They observed that both motion and shape are essential for accurate classification in complex videos. Laptev *et al* [2] then extended his previous work [23] to classify 8 complex natural actions found within Hollywood movie films including *Answerphone*, *GetOutCar* and *Kiss*. Multiple scales are used to extract volumes centred over detected interest points. Each volume was subdivided into a number of cuboids, and in each cuboid, HoG and HoF features are computed and concatenated. A bag of spatio-temporal feature words was then built and a non-linear SVM used for classification. The use of a volume indicates the importance of the spatio-temporal relationship between the features.

A further idea that is being exploited to achieve success on complicated datasets is that of identifying context. Han *et al* [27] and Marszalek *et al* [14] learn the context of the environment in addition to the actual action. Han applies object recognition to learn relationships such as the number of objects and distance between them, in order to boost a standard SIFT based HoF/HoG [2] bag of words approach. Marszalek *et al* [14] builds on the previous work by Laptev [2] by learning the
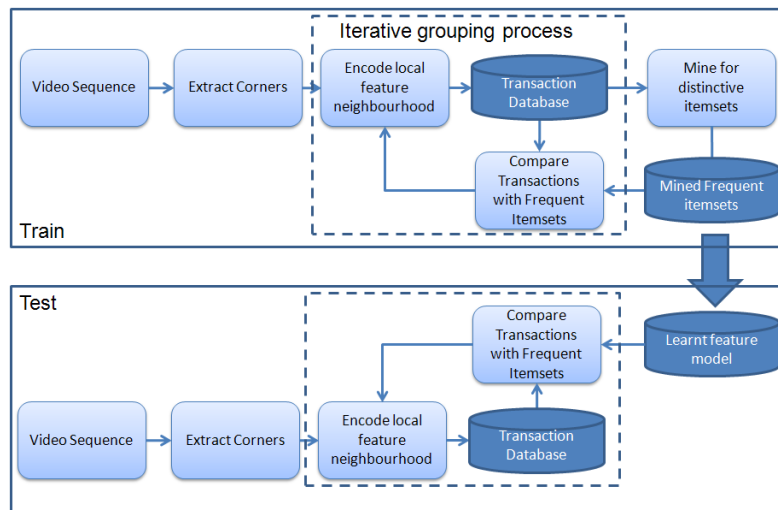
Fig. 3. Overview of Approach

context in which actions occur. They use the intuition that certain actions will only happen in specific scenes, for example *GetOutCar* will occur only in scenes labelled as *Outdoors* or *InCar*. Therefore, by detecting the scene in which the action is occurring, the action classification can be improved. The scene model is learnt using 2D Harris corners with SIFT descriptors, while using the HoF and HoG descriptors of Laptev [2] to recognise the action. The addition of the scene information allows for an accuracy increase of between 1% and 10% per action. At the other extreme Wang *et al* [8] compare different traditional sparse interest point detectors with dense sampling of feature points and find that dense sampling outperforms all other approaches for realistic datasets. However, they note that the very large number of features can be difficult to handle compared to the sparsely engineered interest point detectors.

The scale of the datasets in temporal based action recognition directly lend themselves to data mining algorithms, especially where only weak supervision is available. However, most previous applications of mining have been within the imaging field. Tesic *et al* [28] used a data mining approach to find the spatial associations between classes of texture from aerial photos. Similarly Ding *et al* [29] derive association rules on Remote Sensed Imagery data using a Peano Count Tree (P-tree) structure with an extension of the more common *APriori* [30] algorithm. Chum *et al* [31] use data mining to find near duplicate images within a database of photographs, while Quack *et al* [10] applied *Association rule* data mining to object recognition by mining spatially grouped SIFT descriptors. Yuan *et al* [32] use frequent itemset mining to first select the weak classifiers for Adaboost to the applied to, they argue that this method removes poor classifiers that could reduce the accuracy even after boosting.

In summary in can be seen that most published approaches use a two state process, involving feature detection and description, followed by classification. In contrast our proposed method combines the feature selection and classification into a single approach.

3 APPROACH OVERVIEW

Figure 3 shows an overview of the approach for training and testing. Initially, 2D corners are detected in three orthogonal planes of the video sequence (x, y), (x, t) and (y, t). There can be over 1,500 corners per frame, presenting an over complete set of features with large amounts of redundancy and noise. Each corner is encoded as a three digit number denoting the spatio-temporal plane in which it was detected, the scale at which it was detected and it's orientation. These corners are then used within an iterative hierarchical grouping process to form descriptive compound features. Each corner is grouped within a cuboid based neighbourhood. A set of grouped corners is called a *Transaction* and these are collected to form a *Transaction* database. This database is then mined with the purpose of finding the most frequently occurring patterns. These patterns are descriptive, distinctive sets of corners and are called frequent *itemsets*. The mined frequent *itemsets* then become the basic features for the next level of mining. These compound corners are then grouped within an enlarged spatio-temporal neighbourhood to form a new Transaction database on which data mining (search for frequently occurring substrings) can again be performed. The process is iterated with the final stage Frequent Itemsets becoming the class feature model. For classification of unseen data, the process is identical apart from the final iterative loop where compound features are compared to the model learnt in the training phase. A voting mechanism is used to score detected Itemsets against learnt/mined models. Finally, as the Frequent Itemset encoding contains information of every constituent corner location, a pixel based likelihood image for each action can be accumulated, allowing localisation to be performed.

4 DATA MINING

Data mining allows large amounts of data to be processed to identify any reoccurring patterns within the data in a computationally efficient manner. One mining algorithm is Association rule [33] mining. This was originally developed for supermarkets to analyse shopping bought by customers, with the aim of finding regularity in the shopping behaviour of those customers. The aim was to find *association rules* within millions of shopping Transactions. An association rule, Λ, is a relationship of the form $\{A, B\} \Rightarrow C$, where A, B and C are sets of items. A and B are the antecedent and C the consequence. An example of the rule might be, customers who purchase items A and B are very likely to purchase items C at the same time. The belief in each rule is measured by a support and a confidence value.

To process the Transaction association rules, Agrawal [30] developed the *APriori* algorithm. It can be formulated in the following way, if $I = \{i_1, ..., i_p\}$ is a complete set of p discrete items then 2^p subsets can be constructed using the items. Formally, these subsets are the elements of the Power Set of I, $\mathcal{P}(I)$ with cardinality $|\mathcal{P}(I)| = 2^p$ and each set $T \in \mathcal{P}(I)$ is known as an itemset. However, in any particular application, only a limited number of itemsets T_i, known as Transactions, will be observed. The list of observed Transactions form a Transaction database, $D = \{T_1, ..., T_n\}$. The purpose of the Apriori algorithm is to search this database and determine the most frequently occurring itemsets.

As a specific example, consider the set of items $I = \{a, b, c, d, e\}$. There are a possible $|\mathcal{P}(I)| = 2^5$ itemsets but in a specific application, only some of these will be observed. For example, only five itemsets might occur in practice, yielding the following Transaction database, $D = \{\{a, b, c\}, \{a, b, d, e\}, \{a, b, e\}, \{a, c\}, \{a, b, c, d, e\}\}$ where $|D| = 5$

Note that Transactions are itemsets and can be of varying size. The Apriori algorithm is a generative algorithm that uses a breadth first, bottom-up strategy to explore itemsets of increasing size, starting from single item itemsets and increasing the itemset size by one at each level of the search tree. It evaluates the frequency of occurrence of each generated subset using the observed Transaction database but retains only those itemsets whose frequency exceeds some user specified minimum frequency threshold. The Apriori algorithm exploits the heuristic that if an itemset does not exceed the minimum frequency threshold, then none of its descendants (supersets) at the higher levels of the tree can do so and hence these larger size itemsets need never be generated. This heuristic allows the tree to be pruned to reduce the search space and makes the algorithm efficient.

The frequency of an itemset is related to the support and confidence for an association rule, Λ. An association rule of the form $A \Rightarrow B$ is evaluated by looking at the relative frequency of its antecedent and consequent parts i.e. the itemsets A and B. The support for an itemset measures its statistical significance i.e. the probability that a Transaction contains the itemset. For A, this is calculated as the size of the set of all T, such that T is an element of D and A is a subset of T, normalised by the size of D. Using set builder notation, this can be formalised as

$$sup(A) = \frac{|\{T \mid T \in D, A \subseteq T\}|}{|D|} \in \Re \to [0, 1) \quad (1)$$

The support of the rule $A \Rightarrow B$ is therefore

$$sup(A \Rightarrow B) = \frac{|\{T \mid T \in D, (A \cup B) \subseteq T\}|}{|D|} \quad (2)$$

and measures the statistical significance of the rule. The confidence of a rule is then calculated as

$$conf(A \Rightarrow B) = \frac{sup(A \cup B)}{sup(A)} = \frac{|\{T \mid T \in D, (A \cup B) \subseteq T\}|}{|\{T \mid T \in D, A \subseteq T\}|} \quad (3)$$

The support for the rule is the probability of the joint occurrence of A and B i.e. $P(A, B)$ while confidence is the conditional probability $P(B|A)$.

For example, if we consider the association rule $\{a, b\} \Rightarrow c$ in the example Transaction database D given above, then the support of the itemset $\{a, b\}$ is 0.8 i.e. four occurrences of $\{a, b\}$ in five Transactions, while the confidence of the rule is 0.5 i.e. two occurrences of $\{a, b, c\}$ in the four Transactions that contain $\{a, b\}$.

In action recognition, we are interested not solely in the frequency of feature configurations but additionally require them to be discriminatory. To achieve this, the algorithm is run on datasets comprising of both positive and negative examples. The Transaction vectors of all examples are appended with an action label, α, that identifies the class that it belongs to. The results of data mining then include rules of the form $\{A, B\} \Rightarrow \alpha$ and an estimate of $P(\alpha|A, B)$ is given by the confidence of the rule. As the Transaction database contains both positive and negative training examples $P(\alpha|A, B)$ will be large only if $\{A, B\}$ occurs frequently in the positive examples but infrequently in the negative examples. If $\{A, B\}$ occurs frequently in both positive and negative examples i.e. several classes, then $P(\alpha|A, B)$ will remain small as the denominator in the conditional probability will be large.

Ideally all generated association rules would be maintained and the confidence would be used as a measure of discrimination to other action classes. However, due to the sheer number of rules this would be computationally infeasible, therefore both support and confidence are used to filter generated rules. A single support value is used throughout all the stages of mining and is determined as the lowest value that is computationally feasible at the initial level. During mining, only association rules Λ that pass the minimum support criteria T_{supp} are retained. Each generated association rule, Λ, that contains a class label is considered to be a distinctive

feature of the class if its confidence value is above a user specified threshold

$$conf(\Lambda \Rightarrow \alpha) > T_{conf} \qquad (4)$$

It is therefore added to a list called the Frequent Mined Configuration vector for that class, $M(\alpha) = \{\Lambda_1, ..., \Lambda_N\}$ for the N highest confidence association rules. This process is used to mine sets of grouped features in l levels of a hierarchy providing $M^l(\alpha)$. T_{conf} is set to the reciprocal of the number of classes, as this has proven to deliver balanced transaction databases in our experiments.

5 FEATURES

In our work we use dense 2D Harris corners [34]. Laptev *et al* [35] proposed 3D corners as simple features in (x, y, t). However, Laptev's 3D corners are sparse, so instead for our work, we detect 2D corners independently in each of 3 orthogonal planes of the video volume i.e. the gradient interest points are found independently in (x, y), (x, t) and (y, t). This provides information on spatial and temporal image changes but results in a much denser representation than full 3D Harris corners [35], [2]. Interest points are extracted at multiple scales. If i indicates scale then we use search windows of size $\sigma_i = 3 \times 2^{i-1}$, with $i = 1, ...5$ viz $3 \times 3, 6 \times 6, 12 \times 12, 24 \times 24$ and 48×48. This range is sufficient for video sizes up to 640x480 pixels. For larger image sizes further scales could be easily incorporated. In practice, rather than use larger size scanning windows, we create image pyramids by successive 2×2 block averaging of a base image and then use a fixed 3×3 window to find interest points at each level. Figure 4 shows an example of corner detections on three frames taken from various test datasets. The number of detected corners varies depending on the scene. Figure 4(a) from the simpler *KTH* dataset with a uniform background shows few detections outside the action itself, while Figure 4(b) from the *Multi-KTH* dataset has a large number of corners ($\tilde{1}$,500 per window) on the cluttered background. The *Multi-KTH* data involves a moving camera which often results in (x, t) and (y, t) corners firing on background clutter. The same also occurs on the *Hollywood* dataset in Figure 4(c). Also seen in this figure, are a large number of corners firing in (x,y) due to compression artefacts from MPEG encoding. This large number of corners would be unsuitable for many learning methods used in action recognition. However, the hierarchical neighbourhood grouping and data mining is capable of handling these extremely large features sets.

To characterise the interest points, the dominant orientation of the corners is also recorded. This is quantised into k discrete orientations. In our experiments $k = 8$, therefore the orientation is quantised into bins of size $\frac{1}{4}\pi$ radians aligned with the points of a compass. Each detected interest point is represented by a 3 digit string encoding $[\Delta Scale, Channel, Orientation]$, with the first digit representing the difference in the scale between the

scale at which the corner was detected and the scale of the reference point (see Sec 6) $Scale = \{1, ..., 5\}$. The second digit indicating the video plane or channel that the interest point was detected in $Channel = \{1, ..., 3\}$ with $1 = (x, y)$, $2 = (x, t)$ and $3 = (y, t)$ and finally the third digit showing the dominant orientation of the corner quantised into one of 8 equal sized bins $Orientation = \{1, ..., 8\}$. Figure 5 gives a visual example of the encoding.
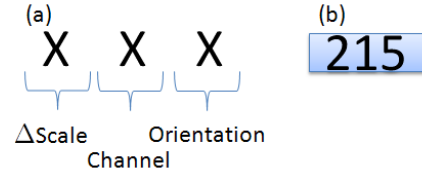

Fig. 5. (a) the three parts that make up a local feature descriptor. (b) this descriptor, is at scale 2 (6x6 patch), dimension 1 (x,y) and corner orientation 5

By detecting and encoding corners in the 3 channels of space and time, we assume a constant relationship between them. This assumption remains valid provided the relationship is preserved. This can easily be archived by ensuring all videos are resampled to a reasonably consistent resolution and frame rate. Exact spatial resampling is undesirable as aspect ratios vary and should be preserved.

6 RECOGNITION FRAMEWORK

In many object and action recognition approaches it has been shown that spatial information can improve accuracy [10], [2]. Individual 2D corners alone have little discriminatory power but consistent spatio-temporal structures formed by grouping several 2D corners are very powerful both for recognition and for rejecting features arising from background clutter.

There are a number of ways to represent the structures in the neighbourhood of a given corner. Sivic [17] simply uses a clustering approach and records the j corners closest to a central corner, without making the spatial relationships explicit. In contrast, Quack *et al* [10] represent the spatial layout of high level SIFT features by quantising the space around a feature using a 2D grid. Each feature is assigned to a cell of the grid. A similar approach is adopted in our method but we use a spatio-temporal hierarchy. The low levels of the hierarchy correspond to structures with a small spatio temporal extent while higher levels associate corners and corner structures over larger scales. At the final stage, the relative location of constituent compound features is encoded in a way that assists scale invariant recognition. Compound features found at higher levels naturally describe more complex structures but at these higher levels there are fewer structures detected. The use of hierarchy both speeds up the classification and leads to
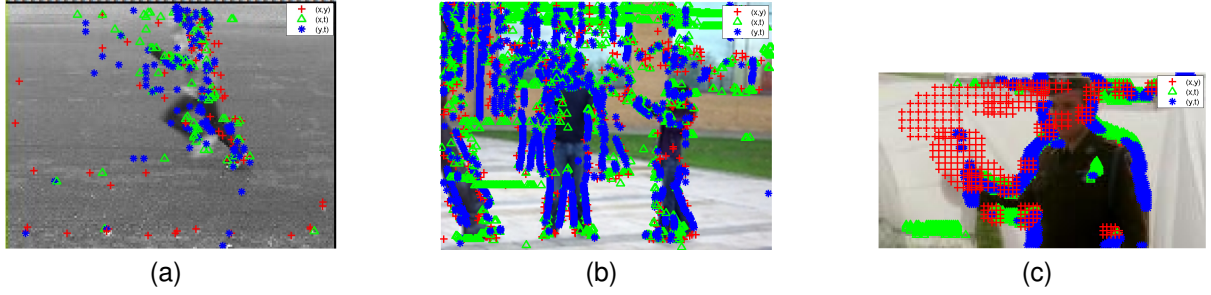
Fig. 4. 2D Harris Corner Detections on Frames, (a) *Running* from the *KTH* dataset [1], (b) *Boxing* and *Handclapping* from *Multi-KTH* Dataset [15], (c) *HugPerson* from the *Hollywood* Dataset [2],

increased accuracy, in a similar way to that of a cascaded classifier [36].

6.1 Neighbourhood Encoding

A regular $3 \times 3 \times 3$ grid, that yields 27 equally sized cells, is used to establish a neighbourhood for encoding the relative position of corners. The grouping is applied at several scales or levels in a hierarchy. At level l of the hierarchy, a cell of the neighbourhood grid extends over ω^l pixels and frames. At the lowest level, $l = 1$, the cell size ω^1 is set to one. At higher levels, cell size is given by $\omega^l = 2 \times \omega^{l-1}$. The hierarchy has up to L levels. In the experiments of Section 7, $L \leq 5$. For $l = 1$, the grid is centred at each 2D corner and other corners that fall within the grid are labelled with a number that denotes the cell that encloses the corner. This provides tolerance to the exact layout of features during encoding. The fact that mining is seeking reoccurring feature combinations, mitigates some of the boundary issues associated with such coarse quantisation.
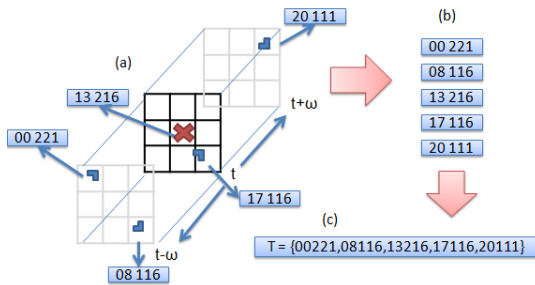

Fig. 6. (a) The grid centred on a corner shown by a cross. Four other corners are found within the neighbourhood defined by the grid. (b)The spatial and temporal encoding of each of the five corners in the grid. (c) The Transaction vector T formed by concatenating the codes for all 5 corners

Figure 6(a) shows four corners that have been identified in the region around a central corner that is marked with a red cross. Each corner has it's individual three digit code based on it's Δscale[1], orientation and direction

---

1. $\Delta Scale$ is taken as the absolute difference between the scale of the central corner and the corner being encoded.

e.g. the centre corner's attributes are given as 216. This code is then pre-appended with an integer that denotes the grid cell where it occurs. For the central corner the cell number is 13 and hence the centre feature is represented by the string 13216. This string is known in data mining as an item and encoding all the corners in the grid yields the 5 items shown in Figure 6(b). The items are then concatenated into a larger 1D vector, known within the mining community as a *Transaction vector* T. Hence, each corner generates a Transaction vector and the ith corner at the 1st level of hierarchy will produce a Transaction vector denoted by T_i^1. Finally, for the purposes of the training stage, each Transaction vector is appended with the label of the associated action class, α. Hence, the Transaction vector from Figure 6 is $\{00221, 08116, 13216, 17116, 20111, \alpha\}$. This encoding process is then repeated for all 2D corners detected in the video sequence to produce D^1, the transaction database for the first stage of mining.

6.2 Learning

The mining process is applied in a hierarchical manner to discover, for each action, α, at each level, l, a set of discriminative frequent mined corner configurations, $M^l(\alpha)$. $M^l(\alpha)$ is a set of itemsets or association rules, Λ, derived from the Transaction database found at level l that frequently occur in the desired action class but are uncommon in other action classes. The elements of $M^l(\alpha)$ identify distinctive configurations and these configurations form an input to the next higher level of hierarchical grouping.

At the second level and above, the features on which mining is performed are no longer simple corners, but compound groupings of corners from the last level of mining. The region encoding therefore only needs to capture the spatio temporal relationship of compound features at the last level, as the concept of scale and orientation no longer exist. $D^{1>l>L} = \{T_i^l\}^{|D^l|}$ where the transaction vector T_i^l is built in terms of the Λ^{l-1} symbols pre-appended with the integer that denotes the grid cell where it occurs as for $l = 1$, e.g. $D^{1<l<L} = \{\{r_1\Lambda_1^{l-1}, \ldots, r_2\Lambda_2^{l-1}\}, \ldots, \{r_3\Lambda_3^{l-1}, \ldots, r_4\Lambda_4^{l-1}\}\}$ where $r_* \in \{0, \ldots, 26\}$ is the grid cell in which a compound

features fires and Λ^{l-1} is the presence of a compound feature found at the previous level of the hierarchy.

6.2.1 Scale Invariant Grouping

At the final level of the hierarchy $l = L$, the grouping is changed from a $3 \times 3 \times 3$ grid to a $2 \times 2 \times 3$ grid centred on the feature (see figure 8). This grid is divided into 12 equal cuboids in the x, y, t domain where x, y radiate from the centre of the neighbourhood out to the image edge, and t extends to successive and preceding frames based on ω_l. This discards the spatial distance between features and simply encodes the relative displacement of features. It assists recognition with invariance to scale and is possible as the features that fire at the higher levels are very sparse and therefore result in a relatively small transaction size despite encoding all features. At previous levels of the hierarchy such an approach becomes infeasible as the number of features in the x, y plane alone are prohibitively large.

Fig. 8. Note for illustration, single frame grounding is shown. The grid is centred upon the feature to encode and extends to the boundary of the image in x,y and a single frame either side in t. (a) There are two compound features found on this frame, these have been prefixed by their grid location in (b). (c) Shows the concatenation of the local features into a Transaction vector for this interest point.
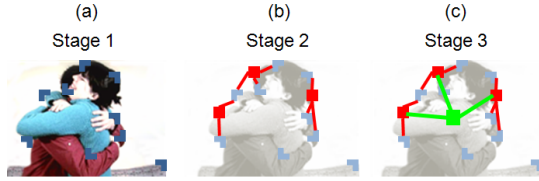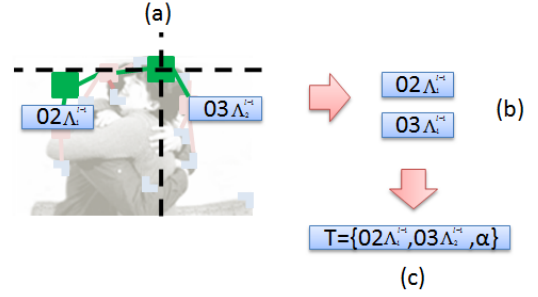

Fig. 7. The hierarchical construction of compound features, (a) shows the initial Harris corners, (b) shows three compound features made up of the corners from (a). (c) shows a single compound feature representing all the corners

Figure 7 illustrates how the initial 2D corners are formed into compound features as the level of hierarchy increases. Figure 7(a) shows some initial 2D detected corner features. Using a grid whose cell size is 1 and which extends over $3 \times 3 \times 3$ pixel/frames around a corner point, a set of Transaction vectors is generated i.e. in the example there are 9 corners and hence 9 Transaction vectors $D^1 = \{T_i^1\}_{i=0}^9$. However, following mining, a set of frequent mined corner configuration corresponding to association rules whose support passes the threshold is generated. In this example the number of rules discovered is 3 so $M^1(\alpha) = \{\Lambda_1^1, \Lambda_2^1, \Lambda_3^1\}$. These are shown in red. At $l = 2$, the grouping size is now increased, $\omega_2 = \omega_1 * 2$ and for each Λ_i^1, all other Λ_j^1 that fall within the local grid are again appended with the grid location to form a new transaction database D^2 for the next level of mining. Following mining on D^2, $M^2(\alpha) = \{\Lambda_1^2\}$ i.e. a single compound feature is constructed indicated as the green hierarchical constellation of corners in Figure 7c.

6.3 Recognition

Once the training has occurred, the frequently reoccurring distinctive and descriptive compound features for each class, α are produced, $M(\alpha) = \{M^l\}_1^L$. To classify an unseen video sequence, it is analysed in a similar fashion

to the approach outlined during learning. The iterative process of encoding features in terms of $M(\alpha)$ symbols is repeated to form transaction databases, but instead of mining patterns from D, only patterns that exist in $M(\alpha)$ is passed to the next level. The confidence of each transaction in $M(\alpha)$ are used to weight the matches, as a high confidence would indicate that the Transaction, T is distinctive compared to other classes. The use of the confidence ensures that if the transaction is matched with several classes, the confidence will provide a measure of the discrimination between those classes. The response, R of the classifier is given by

$$R_\alpha = \frac{1}{|D \cap M(\alpha)||M(\alpha)|} \sum^{\forall T_i \in D} m(T_i, M(\alpha)) \quad (5)$$

where

$$m(T_i, M(\alpha)) = \begin{cases} conf(T_i \Rightarrow \alpha) & T_i \in M(\alpha) \\ 0 & otherwise \end{cases} \quad (6)$$

The model score can then be used in several ways to make a decision about the action class of the video sequence. We have chosen to accumulate responses for each frame of the sequence, assign an action label according to the class that maximises the response for that frame and then take a majority decision over all frames to decide the action label for the complete video sequence. In the unlikely event that no matches occur and the model score is zero, the video would be classed as not containing any action.

6.4 Localisation

For video sequences where a single action occurs, the classification process outlined above is sufficient. However, if multiple actions occur simultaneously then localisation of each action will be required. To achieve

this we use the frequently reoccurring compound feature for each class, $M(\alpha)$ to generate a confidence-based likelihood map for all locations in the sequence's space-time volume. Each feature that fires within an image is encoded at a fixed scale and as such, knows the area of influence the video had upon it. Each feature which is within $M(\alpha)$ can therefore vote for the area of the image in which the action is occurring. The likelihood response for the entire image is maintained in the form of an accumulator array which is initially set to zero. When a feature fires, the response value of equation 5 is added to the appropriate region of the accumulator around the position of the central feature. Figure 16(a) shows the likelihood image for a typical example. A threshold can then be applied on a pixel by pixel basis to find the most likely locations for a given action. This is shown in Figure 15 for frames from the *Multi-KTH* dataset with the most likely positions of the actions colour coded. It should be noted that the localisation highlights the action and not the person performing the action. Hence, in these examples, the activity is centred on indicative motion (i.e. hands and legs) rather than the person.

7 EXPERIMENTAL RESULTS

Four different datasets were used to test the approach proposed within this paper. The focus of each dataset is different, to illustrate the generalisation of the method. The datasets used include the well known and popular *KTH* dataset [1], to provide a comparison with the other techniques reported in the literature. The simultaneous multi-action *Multi-KTH* dataset [15], using the same actions and training as the *KTH* dataset, demonstrates detection of multiple actions in noisy scenes with background clutter and a moving camera. Finally, two natural real-world datasets are used, *Hollywood* [2] and *Hollywood2* [14], both are made up of clips from movie films. On all datasets our approach outperforms competing state-of-the-art approaches reported in the literature.

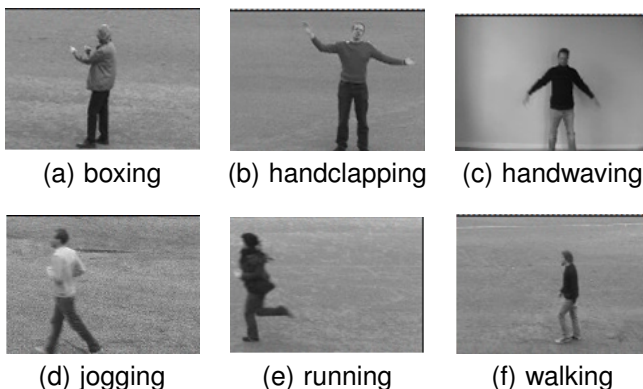The *KTH* dataset contains 6 different actions; *boxing, hand-waving, hand-clapping, jogging, running* and *walking*, examples of each action are shown in Figure 9. The state-of-the-art recognition accuracy on the *KTH* dataset are generally within the range of 86% to 95% and therefore there is little room for improvement. To provide an additional challenge, the *Multi-KTH* dataset [15] was proposed. It consists of a single 753 frame long sequence where multiple people perform the *KTH* actions simultaneously. To increase difficulty, there are large changes in scale, camera motions and a non uniform background. Some frames from the sequence are shown in Figure 10. The third action recognition dataset is

Fig. 10. Examples from the *Multi-KTH* dataset

the *Hollywood* dataset of Laptev [2]. It consists of 8 actions; *AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp, StandUp*, with clips taken from Hollywood films, see Figure 11. The fourth dataset is
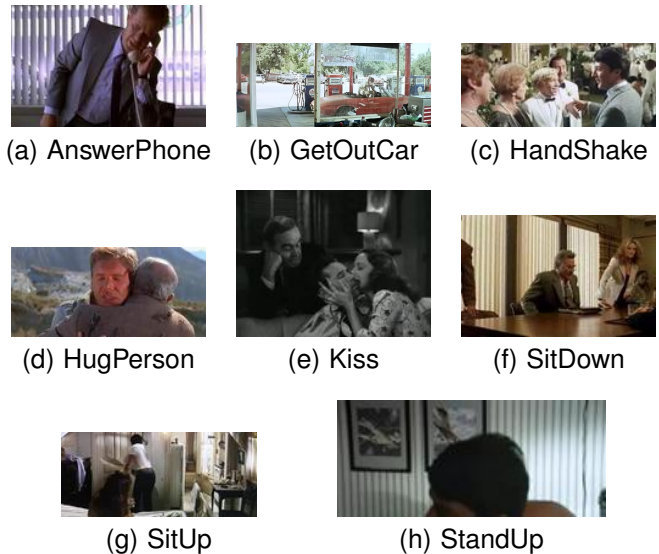
Fig. 11. Examples from the *Hollywood* dataset [2]

the *Hollywood2* dataset [14]. It builds upon [14] and consists of 12 action classes; *AnswerPhone, DriveCar, Eat, FightPerson, GetOutCar, HandShake, HugPerson, Kiss, Run, SitDown, SitUp, StandUp* with around 600,000 frames or 7 hours of video sequences split evenly between training and test datasets.

7.1 Performance Measures

For the *KTH* dataset, the data for training and testing can be partitioned in two ways. The partition originally proposed by the dataset's creator Schüldt [1] is used to allow direct comparison of results. There are a total of 25 people performing each action 4 times, giving 599 video

Fig. 9. Examples from the *KTH* dataset

sequences (1 sequence is corrupt). Each video contains 4 instances of the action totalling 2396 unique actions. We present results using training and testing data split as suggested by Schüldt, with 8 people for training, and 8 people testing. However, many authors ignore this partitioning and instead opt for an easier leave-one-out cross validation. While this is a far simpler test, we also report results in this fashion to allow comparison with other methods.

For the *Multi-KTH*, accuracy of the localisation is used as the measure of performance. Each action has a manually groundtruthed rectangular bounding box and the action localisation is deemed correct if the resultant dominant pixel label within the bounding box matches the ground truth. Visual examples of the localisation are shown in Figure 15 and Figure 16.

The clean test and training partitions proposed by Laptev [2] were used for the *Hollywood* dataset to allow direct comparison to their published results. There are 219 training video sequences spread over the 8 actions, and 211 test video sequences. For the *Hollywood2* dataset, the clean train and test partitions proposed by Marszalek [14] were used. There are a total of 810 training videos spread over 12 action classes, with 884 test sequences. An important point to note is that for all datasets, none of the movies used in training are used in the test sequences, meaning the classifier's aren't trained on the film but on the actual human action. Each of the datasets offers different challenges therefore they will be examined in turn, beginning with the popular *KTH* dataset.

7.2 *KTH* dataset Action Classification

While the *KTH* dataset is generally seen as simplistic due to the near uniform background and artificial actions performed, it is useful to compare with other state-of-the-art methods. Figure 12 shows the resulting average precision confusion matrix for the six actions of the *KTH* data using for the Schüldt training / test partition [1]. There is most confusion occurring between

	box	clap	wave	jog	Run	Walk
Box	100	0	0	0	0	0
Clap	0	94	6	0	0	0
Wave	0	1	99	0	0	0
Jog	0	0	0	91	7	2
Run	0	0	0	10	89	1
Walk	0	0	0	0	6	94

Fig. 12. Confusion matrix of human action recognition results for *KTH* dataset using training/test partition proposed by [1]

jogging and *running* and between *handClapping* and *handWaving*, which are common confusions reported for competing techniques. Table 1 shows the average precision compared to other state-of-the-art approaches for the Schüldt training / test partition.

TABLE 1
Average precision on *KTH* action recognition Dataset using **training/test partition proposed by [1]**

Method	Average
Schüldt training/test partitions	Precision
Wang *et al* [8] Harris3D + HOF	92.1%
Laptev *et al* [2] HOG + HOF	91.8%
Klaser *et al* [37] HOG3D	91.4%
Nowozin *et al* [38] Subseq Boost SVM	87.04%
Schüldt *et al* [1] SVM Split	71.71%
Ke *et al* [24] Vol Boost	62.97%
Fixed grid	88.5%
Non-Hierarchical Mined, $L = 1$	89.8%
Hierarchical Mined, $L = 3$	94.50%

The table shows our Hierarchical Mined Approach technique has higher classification accuracy than all other state-of-the-art methods. This includes the various feature descriptor combinations of **HOF** and **HOG** from Wang [8] and Laptev [2] and **Subseq Boost** the boosted SVM classifier by Nowozin [38].

The use of a hierarchical approach provides a 5% increase over no hierarchy and provides a good increase in performance over previously published results. Results within Table 2 use the simpler leave-one-out approach. This shows higher overall average performance, however, our approach still out-performs all other approaches and gives a comparison of the complexity of leave-one-out cross validation versus the training/test split of [1].

TABLE 2
Average precision on *KTH* action recognition Dataset using **leave-one-out cross validation**

Method	Average
leave-one-out test/train	Precision
Kim *et al* [39] CCA	95%
Zhang *et al* [40] BEL	94.33%
Liu and Shah [41] Cuboids	94.15%
Han *et al* citeHanICCV09 MKGPC	94.1%
Uemura *et al* [15] Motion Comp Feats	93.7%
Bregonzio *et al* [42] 2D Gabor filter	93.2%
Yang *et al* [43] Motion Edges	87.3%
Wong and Cipolla [44] Subspace SVM	86.60%
Niebles *et al* [45] pLSA model	81.50%
Dollar *et al* [20] Spat-Temp	81.20%
Fixed grid	90.5%
Non-Hierarchical Mined, $L = 1$	91.7%
Hierarchical Mined, $L = 3$	95.7%

Table 3 shows the average precision of performance over five stages of the Hierarchical grouping for the Schüldt partition. Initially as the stages increase and the compound features become more complex, the accuracy increases. This is because the compound features become

TABLE 3
Average Precision over the hierarchical stages on a per action basis on *KTH* dataset

Action	Hierarchy stage				
	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$
Boxing	93%	93%	100%	91%	80%
HandClapping	84%	90%	94%	82%	64%
HandWaving	92%	91%	99%	90%	55%
Jogging	87%	91%	91%	91%	81%
Running	87%	88%	89%	86%	74%
Walking	96%	96%	94%	94%	84%
Average Precision	89.8%	91.5%	94.5%	89.0%	73%

more complex at each level of the hierarchy and are therefore able to differentiate more reliably between different action classes. $L = 1$ is the non-hierarchical approach where a single grouping stage is performed using the $2 \times 2 \times 3$ encoding. $L = 2$ is a two stage grouping where $3 \times 3 \times 3$ is used initially followed by $2 \times 2 \times 3$. Three further stages of grouping are shown with $L = 3$ providing the optimum performance. However, by the fourth and fifth stages there are too few features to effectively classify, causing an overall reduction in accuracy. However stage 4's performance is still greater than that of all other methods in Table 1.

Figure 13 shows two further confusion matrices for the *KTH* dataset. Figure 13(a) shows the results for a single stage approach using a *fixed* size grid size of $4 \times 4 \times 2$ where $\omega = 15$. It has an average precision of 88.5%. Figure 13(b) shows results for a single stage, scale invariant approach ($L = 1$) which uses the $2 \times 2 \times 3$ encoding to capture the relative displacement from the centre feature as illustrated in Figure 8, see section 6.2.1 for associated discussion of scale invariance.

The scale invariant approach (Figure 13(b)) has an average precision of 89.8%, an increase of just over 1% compared to the fixed grid. This increase is due to the additional invariance to scale gained by the relative encoding and the addition of temporally adjacent features. This small increment in performance is largely due to the ceiling performance of the dataset having been reached.

7.3 Computation Cost

Important points that are often neglected within action recognition, are speed and computational cost of the methods proposed. One of the advantages of using a data mining technique, is the speed of learning patterns when compared to other machine learning approaches such as Boosting or SVMs. In addition, simple 2D corner detection has a relatively low computational cost. The spatial neighbourhood grouping is fast during training as it has limited neighbourhoods in which to encode features. At each level of the hierarchy, the features are grouped into more complex and discriminative features. By using a hierarchical approach, a lower overall computational cost can be achieved compared to using a single level of feature with equivalent complexity. This is due

to the removal of unused compound features at each level of the hierarchy, resulting in less computational overhead. Table 4 shows the average frame rate for the stages of training for the *KTH* dataset. It indicates the real-time nature of the training process, despite being an unoptimised C++ implementation running on a standard single core desktop PC. It also shows that the number of features are greatly reduced by the successive stages of mining and it is this that allows the overall classifier speed to be maintained despite the additional levels of complexity. It should be noted that although the number of features reduces drastically at each level of the hierarchy, the encoding at all previous levels still needs to be performed. Hence the overall reduction in speed at each level. Table 5 shows the average runtime frame rate at testing for the four different datasets used. The table also shows that there is a large variation in frame rate over the datasets. The high frame rate within the *KTH* dataset is due to the simple uniform background reducing the number of features that are detected. In contrast, the data for *Hollywood*, *Hollywood2* and *Multi-KTH* indicate more realistic speeds for a cluttered background and large images where there are a greater number of features to be encoded and grouped.

TABLE 4
A break down of the average frame per second of the successive training stages on the KTH dataset

Stage	Frames per second	Ave features per frame
Encoding l=1	35fps	1500
Mining l=1	640fps	
Encoding l=2	28fps	300
Mining l=2	21fps	
Encoding l=3	18fps	210
Mining l=3	10fps	
Encoding l=4	8fps	30
Mining l=4	8fps	
Encoding l=5	7fps	25
Mining l=5	2fps	

TABLE 5
The average frame rate at runtime for the four datasets

Dataset	Level	Frames per second	Resolution
KTH	3	24fps	160x120
Multi-KTH	3	4fps	320x240
Hollywood	2	10fps	320x240
Hollywood2	2	7fps	320x240

7.4 *Multi-KTH* dataset

The *Multi-KTH* dataset is a more challenging version of the *KTH* dataset. It has the same 6 actions and training video sequences, but the test sequence consists of multiple simultaneous actions, with significant camera motion and scale changes, and a more cluttered and realistic background. Table 6 shows how the performance increases as the number of hierarchical stages increases,

	box	clap	wave	jog	run	Walk
Box	84	2	14	0	0	0
Clap	1	98	1	0	0	0
Wave	15	0	85	0	0	0
Jog	0	0	0	82	15	3
Run	0	0	0	15	85	0
Walk	0	0	0	3	0	97

(a) Fixed Grid

	box	clap	wave	jog	run	Walk
Box	93	2	0	0	3	1
Clap	14	84	0	1	0	1
Wave	2	0	92	1	0	4
Jog	3	0	0	87	1	6
Run	2	0	0	7	87	3
Walk	0	0	0	0	4	96

(b) Scale Invariant

Fig. 13. Confusion Matrix of Precision for *KTH* dataset

and we see a similar peak performance at stage 3 as was seen in Table 3

TABLE 6
Average Precision over the stages on *Multi-KTH* dataset

Action	Hierarchy stage				
	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$
Boxing	76%	76%	75%	72%	34%
HandClapping	75%	65%	69%	70%	45%
HandWaving	84%	80%	77%	76%	61%
Jogging	50%	61%	85%	51%	24%
Walking	59%	61%	70%	60%	38%
Average Precision	68.8%	68.6%	75.2%	65.8%	40.4%
Average Recall	64.3%	70.2%	74.3%	60.1%	29.3%

The table shows the increase in performance from a $L = 1$ classifier with no hierarchical compound features, to a peak performance using a three stage hierarchical classifier of 75.2%. By the fourth and fifth stages, the performance starts to decrease, this is due to too few compound features firing. Table 7 gives the results over
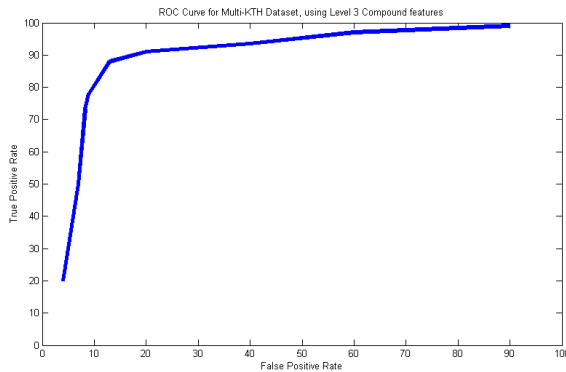

Fig. 14. ROC Curve for the Level 3 compound features on the MultiKTH Dataset

the 5 actions compared to previously published results from Uemura. [15].

Our stage 3 (L3) hierarchy approach has a significant increase in accuracy. The use of the hierarchy allows for

TABLE 7
Average precision of actions on *Multi-KTH* action recognition Dataset

	Clap	Wave	Box	Jog	Walk	Ave
Uemura [15]	76%	81%	58%	51%	61%	65.4%
$L = 3$	69%	77%	75%	85%	70%	75.2%

complex compound corner features to be mined, these are then more invariant to the cluttered background and motion of the video sequence especially with dynamic actions such as *walking* and *jogging*. This ensures a higher true positive rate, while reducing the false positive detections on the background. Figure 14 shows the ROC curve for the Level 3 Compound Features. As can be seen, the results of Table 7 are achieved with a false positive rate (FPR) of <9%. Table 8 shows the effect on the average frame per second as the number of hierarchical stages increases. Figure 15 shows four example frames of action

TABLE 8
Average Frame per second over the hierarchical stages on *Multi-KTH*

Hierarchy stage	Ave fps
$L = 1$	0.03 fps
$L = 2$	1 fps
$L = 3$	4 fps
$L = 4$	15 fps
$L = 5$	16 fps

localisation.

It can be seen that the localisation is generally centred on the person's upper body and hands for the static actions (*boxing*, *handClapping* and *handWaving*) and is centred around the legs for the dynamic actions. This is because the legs contain the descriptive features for the dynamic actions, while the localisation of the hands and body is important for the static actions. These results are impressive as no ground truth training data was provided, only the class labels of the video. In comparison, the approach by Uemura required ground truth positions during training. Figure 15(b) is expanded to
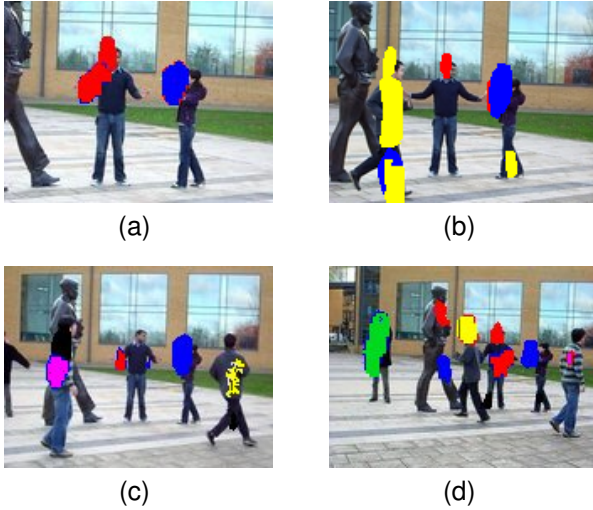
Fig. 15. Localisation results from the *Multi-KTH* dataset, red - *handclapping*, blue - *boxing*, yellow - *running* pink - *walking*, green - *handwaving*

(a) *Boxing*   (b) *HandClapping*
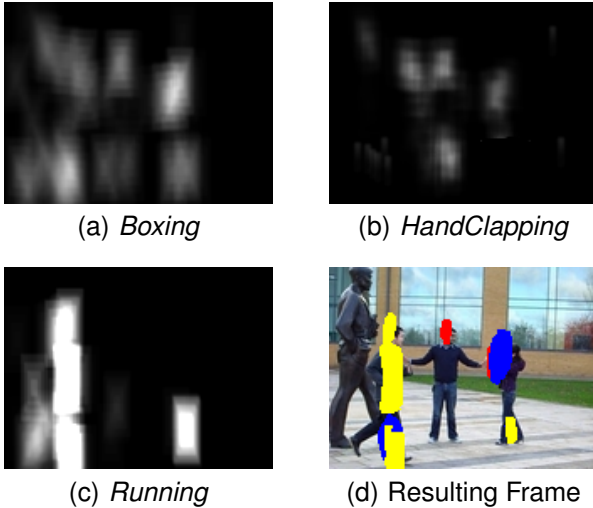
(c) *Running*   (d) Resulting Frame

Fig. 16. Localisation results from the *Multi-KTH* dataset

show the separate action likelihoods in Figure 16, using a common normalized scale. Figure 16(d) is produced by thresholding each likelihood image and taking a pixel by pixel maximum likelihood decision of the action within that area. The threshold is the same as that used for Table 7 and roughly translates to a 9% FPR.

The *Multi-KTH* sequence uses the class models trained on the *KTH* dataset. Comparing the overall average precision with the original *KTH*, an accuracy decrease of around 20%, demonstrates the more challenging nature of the *Multi-KTH* data.

7.5 *Hollywood* dataset

Training was performed with the "clean, manual" dataset of 219 videos and performance was evaluated using the "clean" test dataset of 217 videos. This is the same as reported by Laptev in [2] to allow for a

direct comparison of results. Table 9 presents the average

TABLE 9
Average precision for the *Hollywood* Test Dataset

Action	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$
AnswerPhone	3.1%	25.7%	47.0%	21.5%	2%
GetOutCar	4.5%	38.5%	47.0%	38.4%	32%
HandShake	2.3%	45.6%	50.0%	38.0%	5%
HugPerson	8.6%	42.8%	42.1%	12.3%	0%
Kiss	43.3%	72.5%	69.4%	56.2%	15%
SitDown	28.6%	84.6%	46.2%	25.8%	0%
SitUp	10.2%	29.4%	44.0%	34.4%	0%
StandUp	5.5%	41.6%	70.5%	61.1%	21%
Average	13.2%	53.5%	52.0%	36.0%	9%

accuracy for the 8 actions for a two, three, four and five stage hierarchy, as well as a single stage (non-hierarchical approach $L = 1$). The values are relatively low compared to the *KTH* dataset, however, as Figure 17 shows with examples from the *SitUp* class, there are dramatic illumination, people, and camera angle variations in the data. There is little difference between the stage

(a)   (b)

(c)   (d)

Fig. 17. (a,b) SitUp Sequence 1, (c,d) SitUp Sequence 2

2 and stage 3 hierarchical grouping. However, by the fourth and fifth stages the complexity of the compound features increases, meaning they fire less often. This reduces the feature set and less features mean less ability to discriminate between classes therefore reducing the overall accuracy of the recognition. This trend is similar to the previous two datasets. However, the peak performance is at a lower level than the previous datasets due to a greater variability in actions. To illustrate the reduction in the number of features firing at higher stages, Figure 18 shows the different stages of compound features on a single example frame of a handshake video, and Table 10 gives the actual number of features per frame at each stage. Notice how the features localise the hands at each stage. Also note how features have been selected on the head region. In many training examples of handshake the hands cannot be seen in the frame. Mining has therefore selected additional feature indicative of this class such as the subtle motion of a head nod that accompanies the handshake.
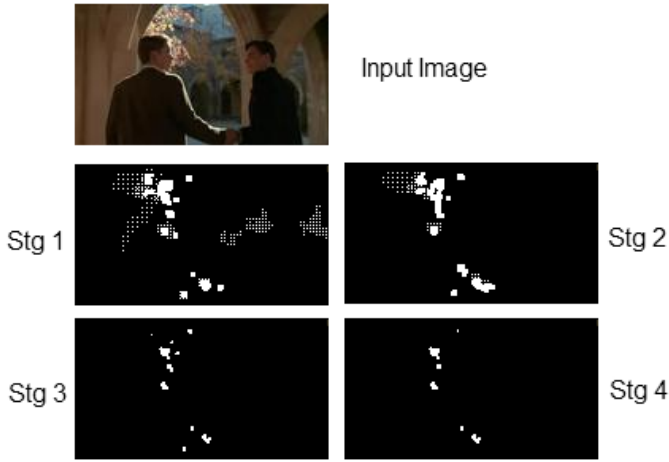
Fig. 18. Detected compound features at each stage for a *handshake*,

TABLE 10
Average Number of features per frame at each stage

Stage	Num of Features
$l = 1$	1214
$l = 2$	1144
$l = 3$	92
$l = L = 5$	61

7.6 Choosing Compound Feature Complexity

Table 9 simply computes average classifier performance for each level of the hierarchy. However, although $l = 2$ provides the highest average recognition rate for this dataset, some classes are more accurately classified at other levels of the hierarchy. This is to be expected, as different actions exhibit varying levels of complexity. Simply choosing the highest accuracy for each row of table 9 would amount to optimising our approach over the test data which is poor scientific practice. Ideally training, test and validation sets would be used but repartitioning of the data would make any comparison to other techniques meaningless. Instead we opt to evaluate accuracy of the hierarchy at re-classifying the training data in the hope that peak performance on training will generalise to peak performance on unseen test data.

The classifier performance was computed on the training data, over the five levels, these results are shown in Table 11. For each action, the level that produced the highest precision was noted and is indicated by the bold type face. Using the peak performance on training to select the appropriate level of the hierarchy specific to each action provides the average precision shown in Table 12. While this approach does not provide optimal results that would be gained from optimising over the test set, it provides a fairer approach and still produces around a 3% increase to give an average precision of 56.8%, which compares very favourably to the previously published

results of 38% and 47.5%.

TABLE 11
Average precision for the *Hollywood* Training Dataset, numbers in **bold** indicate highest AP for action

Action	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$
AnswerPhone	12%	63%	**73%**	15%	4%
GetOutCar	41%	42%	**58%**	31%	8%
HandShake	25%	**54%**	31%	12%	1%
HugPerson	48%	**74%**	44%	40%	4%
Kiss	31%	62%	**64%**	50%	34%
SitDown	28%	**74%**	34%	14%	10%
SitUp	8%	74%	**83%**	47%	14%
StandUp	25%	41%	**72%**	51%	45%

7.7 Mined Transaction size

The combination of individual features is one of the main novelties of the approach, therefore, further analysis of it's actual importance is performed. To present the importance of the length of the mined association rule vectors, the *Hollywood* dataset was tested again using a stage 1, 2 and 3 neighbourhood grouping, but with the maximum length of the association rule vector limited. Table 13 shows the results as the maximum permissible length of items within the mined Transaction vectors is increased and Figure 19 shows a visual representation of these results. The increase in accuracy as the minimum

TABLE 13
Average precision of *Hollywood* dataset, increasing the maximum number of items within the of Transaction vector.

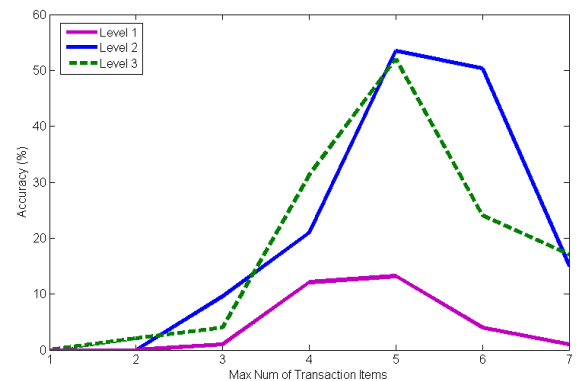| Max No. of Items	2	3	4	5	6	7
L1 Accuracy	0%	1%	12.1%	13.2%	4%	1%
L2 Accuracy	0%	9.6%	21%	53.5%	50.3%	15%
L3 Accuracy	2.1%	4.0%	31.3%	52%	24%	17%
Min fps (s)	0.0001	0.002	0.2	3.5	4.5	30
Max fps (s)	0.001	0.1	2	28	28.5	64

Fig. 19. *Hollywood* Accuracy as maximum number of Transaction items increases

item size increases, indicates that the greater the size

TABLE 12
Average precision for the *Hollywood* Test Dataset, compared with the previously published results.

Action	Laptev [2]	Willems [46]	Matikainen [47]	Han [27]	Ours
AnswerPhone	32.1%	22.9%	0%	43.4%	47.0%
GetOutCar	41.5%	19.5%	7.7%	46.8%	47.0%
HandShake	32.3%	20.4%	5.3%	44.1%	45.6%
HugPerson	40.6%	17.9%	0%	46.9%	42.8%
Kiss	53.3%	33.8%	71.4%	57.3%	72.5%
SitDown	38.6%	21.8%	4.5%	46.2%	84.6%
SitUp	18.2%	50.2%	11.1%	38.4%	44.0%
StandUp	50.5%	49.8%	69.0%	57.1%	70.5%
Average	38.4%	29.6%	31.1%	47.5%	56.8%

of the compound feature, the more important they are within the classifier. This is intuitively reasonable as the more complex compound features are likely to contain greater discriminative detail for improved between-class disambiguation. This is the same principle as the hierarchy and it can be seen that at $L = 2$ (maximum length of 5 and 6), the accuracy is the same as $L = 3$ (maximum length 5). The slight improvement for stage 2 is due to a greater number of association rules exceeding 5 items in length. This is also the cause of the final reduction in accuracy, as there are too few association rule vectors containing six or more items.

7.8 *Hollywood2* dataset

The *Hollywood2* dataset [14] extends the ideology of the *Hollywood* dataset, with a greater number of actions and additional scene training data. The video sequence split for training and test are the same as proposed by Marszalek to allow direct comparison to his results. The results of the iterative grouping process are shown in Table 14. Our results use the approach outlined in the previous Hollywood section, using the semi-optimal hierarchy stage selection, determined by assessing performance on the training data. The stage of the hierarchy used is shown in brackets. The results from Marszalek [14] and Han [27] use scene and object context enhancement to improve accuracy. However, our approach is able to outperform their published results without context. There is a large variation in the hierarchy level used for the actions. This is because some of the actions such as *Answerphone* or *Handshake* are quite small, and therefore need a very complex set of compound features in order to classify the action over the background noise. Therefore, Level 3 or 4 in the hierarchy produces the best results. In contrast, *FightPerson* and *DriveCar* use more global contextual features and therefore they work with lower level features from level 1. Like us, Wang [8] used dense sampling to obtain similar performance, suggesting that sparse feature representations may not be optimal. However, while the approach by Wang is computationally expensive, our approach can still provide real-time operation, with frame rates ranging over *Hollywood2* video sequences between 5fps and 60fps depending on the complexity of the video i.e. the number of corners detected and encoded.

The optimal number of grouping stages varies and generally the more localised and smaller the action, the higher the stage required to provide good class discrimination. Once again, by the 4th and 5th stage of hierarchical grouping, there are too few compound features being detected and grouped to provide consistent accuracy over all actions.

8 CONCLUSION

This paper has presented an efficient solution to the problem of recognising actions within video sequences. The use of a mined hierarchical grouping of simple corners, means that it is fast and able to form complex discriminative compounds of simple 2D Harris corners. Data mining allows for the use of an over complete feature set in order to efficiently learn the sparse complex compound features. This contrasts with the accepted view of using a feature detector that is engineered to be sparse. Four different datasets have been tested including the complex *Hollywood* and *Hollywood2* datasets of film clips. Also the *Multi-KTH* dataset required multiple action localisation and the *KTH* dataset provides a comparison to other approaches. On all four datasets our approach out performs all other published works. Arguably more complex classifier architectures, such as boosting[32] or Support Vector Machines, could be combined with the mined features. However, the high performance of the relatively simple voting mechanism used within this manuscript demonstrates the strength of the features identified by mining. Future work will investigate forming higher levels of the hierarchy and alternative classification architectures as this is where further accuracy could be gained.

ACKNOWLEDGEMENTS

TABLE 14
Average precision of *Hollywood2* Test Dataset

Action	Marszalek [14]	Han [27]	Wang [8]	Ours (multistage)
AnswerPhone	13.1%	15.57%	-%	40.2%(L=3)
DriveCar	81%	87.01%	-%	75.0%(L=1)
Eat	30.6%	50.93%	-%	51.5%(L=2)
FightPerson	62.5%	73.08%	-%	77.1%(L=1)
GetOutCar	8.6%	27.19%	-%	45.6%(L=3)
HandShake	19.1%	17.17%	-%	28.9%(L=3)
HugPerson	17.0%	27.22%	-%	49.4%(L=2)
Kiss	57.6%	42.91%	-%	56.6%(L=2)
Run	55.5%	66.94%	-%	47.5%(L=3)
SitDown	30.0%	41.61%	-%	62.0%(L=2)
SitUp	17.8%	7.19%	-%	26.8%(L=4)
StandUp	33.5%	48.61%	-%	50.7%(L=3)
Average	35.5%	42.12%	47.7%	50.9%

REFERENCES

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: a Local SVM Approach," *In Proc. of International Conference on Pattern Recognition (ICPR'04)*, vol. III, pp. 32–36, 2004.

[2] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pp. 1–8, 2008.

[3] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan, "Categorizing Nine Visual Classes using Local Appearance Descriptors," *In Proc. of IWLAVS*, 2004.

[4] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'01)*, vol. I, pp. 511–518, 2001.

[5] O. Maron and T. Lozano-Prez, "A framework for multiple-instance learning," in *Advances in Neural Information Processing Systems*. MIT Press, 1998, pp. 570–576.

[6] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional Sift Descriptor and its Application to Action Recognition," in *In Proc. of MULTIMEDIA '07*, 2007, pp. 357–360.

[7] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Int. Journal Comp. Vis.*, vol. 65, pp. 43–72, 2005.

[8] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of Local Spatio-Temporal Features for Action Recognition," *In Proc. of BMVA British Machine Vision Conference (BMVC'09)*, 2009.

[9] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2007.4408924

[10] T. Quack, V. Ferrari, B. Leibe, and L. VanGool, "Efficient Mining of Frequent and Distinctive Feature Configurations," *In Proc. of IEEE International Conference on Computer Vision (ICCV'07)*, 2007.

[11] A. Gilbert, J. Illingworth, and R. Bowden, "Scale Invariant Action Recognition Using Compound Features Mined from Dense Spatio-Temporal Corners," *In Proc. of European Conference on Computer Vision (ECCV'08)*, vol. I, pp. 222–233, 2008.

[12] O. Chum, J. Philbin, and A. Zisserman, "Near Duplicate Image Detection: min-Hash and tf-idf Weighting," *In Proc. of BMVA British Machine Vision Conference (BMVC'08)*, 2008.

[13] A. Gilbert, J. Illingworth, and R. Bowden, "Fast realistic multi-action recognition using mined dense spatio-temporal features," *In Proc. of International Conference on Computer Vision (ICCV'09)*, vol. I, pp. 222–233, 2009.

[14] M. Marszalek, I. Laptev, and C. Schmid, "Actions in Context," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009.

[15] H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature Tracking and Motion Compensation for Action Recognition," *In Proc. of BMVA British Machine Vision Conference (BMVC'08)*, 2008.

[16] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-Local Affine Parts for Object Recognition," *In Proc. of BMVA BMVC'04*, vol. II, pp. 959–968, 2004.

[17] J. Sivic and A. Zisserman, "Video Data Mining using Configurations of Viewpoint Invariant Regions," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR04)*, vol. I, pp. 488–495, 2004.

[18] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *In International Journal of Computer Vision*, vol. 20, pp. 91–110, 2003.

[19] G. Willems, T. Tuytelaars, and L. VanGool, "An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector," *In Proc. of European Conference on Computer Vision (ECCV'08)*, vol. II, pp. 650–663, 2008.

[20] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-temporal Features," *ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks*, pp. 65–72, 2005.

[21] J. C. Niebles and L. Fei-Fei, "A Hierarchical Model of Shape and Appearance for Human Action Classification," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'07)*, 2007.

[22] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," *In IEEE Trans PAMI*, vol. 29, no. 12, pp. 2247–2253, December 2007.

[23] I. Laptev and P. Pérez, "Retrieving Actions in Movies," *In Proc. of IEEE International Conference on Computer Vision (ICCV'07)*, 2007.

[24] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient Visual Event Detection using Volumetric Features," *In Proc. of IEEE International Conference on Computer Vision (ICCV'05)*, 2005.

[25] N. Dalal, B. Triggs, and C. Schmid, "Human Detection using Oriented Histograms of Flow and Apperance," *In Proc. of European Conference on Computer Vision (ECCV'06)*, vol. II, pp. 428–441, 2006.

[26] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *In Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 674–679, 1998.

[27] D. Han, L. Bo, and C. Sminchisescu, "Selection and Context for Action Recognition," *In Proc. of International Conference on Computer Vision (ICCV'09)*, vol. I, pp. 1933–1940, 2009.

[28] J. Tesic, S. Newsam, and B. S. Manjunath, "Mining Image Datasets using Perceptual Association Rules," *Proc. SIAM International Conference on Data Mining, Workshop on Mining Scientific and Engineering Datasets*, p. 7177, 2003.

[29] Q. Ding, Q. Ding, and W. Perrizo, "Association Rule Mining on Remotely Sensed Images Using P-trees," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2002, pp. 66–79.

[30] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases*, 1994, pp. 487–499.

[31] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval," *Proc. IEEE International Conference on Computer Vision (ICCV'07)*, pp. 1–8, 2007.

[32] J. Yuan, J. Luo, and Y. Wu, "Mining compositional features for boosting," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1–8, 2008.

[33] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association

Rules between Sets of Items in Large Databases," in *In Proc. of the 1993 ACM SIGMOD International Conference on Management of Data SIGMOD'93*, 1993, pp. 207–216.

[34] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *In Proc. of Alvey Vision Conference*, pp. 189–192, 1988.

[35] I. Laptev and T. Lindeberg, "Space-time Interest Points," *In Proc. of IEEE International Conference on Computer Vision (ICCV'03)*, pp. 432–439, 2003.

[36] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," *Proc. The 13th Conf on MAchine Learning*, pp. 148–156, 1996.

[37] A. Klaser, M. Marszalek, and C. Schmid, "A Spatio-Temporal Descriptor based on 3D Gradients," *In Proc. of BMVA British Machine Vision Conference (BMVC'08)*, 2008.

[38] S. Nowozin, G. Bakir, and K. Tsuda, "Discriminative Subsequence Mining for Action Classification," *In Proc. of IEEE International Conference on Computer Vision (ICCV'07)*, pp. 1919–1923, 2007.

[39] T. Kim, S. Wong, and R. Cipolla, "Tensor Canonical Correlation Analysis for Action Classification," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pp. 1–8, 2007.

[40] T. Zhang, J. Liu, S. Liu, Y. Ouyang, and H. Lu, "Boosted Exemplar Learning for Human Action Recognition," *In Proc. of Workshop on Video-Oriented Object and Event Classification at ICCV'09*, vol. I, pp. 538–545, 2009.

[41] J. Liu and M. Shah, "Learning Human Actions via Information Maximization," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'08)*, 2008.

[42] M. Bregonzio, S. Gong, and T. Xiang, "Recognising Actions as Clouds of Space-time Interest Points," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009.

[43] M. Yang, F. Lv, W. Xu, K. Yu, and Y. Gong, "Human Action Detection by Boosting Efficent Motion Features," *In Proc. of Workshop on Video-Oriented Object and Event Classification at ICCV'09*, vol. I, pp. 522–529, 2009.

[44] S. F. Wong and R. Cipolla, "Extracting Spatio Temporal Interest Points using Global Information," *In Proc. of IEEE Int. Conf. on Comp. Vis. (ICCV'07)*, 2007.

[45] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories using Spatial-Temporal Words," *In Proc. of BMVA British Machine Vision Conference (BMVC'06)*, vol. III, pp. 1249–1259, 2006.

[46] G. Willems, J. Becker, T. Tuytelaars, and L. VanGool, "Exemplar Based Action Recognition in Video," *In Proc. of BMVA British Machine Vision Conference (BMVC'09)*, 2009.

[47] P. Matikaien, M. Herbert, and R. Sukthankar, "Trajectons: Action Recogntion Through the Motion Analysis of Tracked Features," *In Proc. of Workshop on Video-Oriented Object and Event Classification at ICCV'09*, vol. I, pp. 514–521, 2009.

John Illingworth Following a BSc and DPhil(Oxon) in Physics, John Illingworth started a research career in image processing and computer vision in the early 1980's. He has been a full Professor for the last decade and has had substantial funding from both Government and commercial sources. He has co-authored nearly 200 publications, including several prize winning papers. His major research areas have been in feature extraction, shape analysis and 3D data processing. He has been active in community activities and is a former Chairman of the British Machine Vision Association and an editor of two international journals.

Richard Bowden Richard Bowden received a BSc degree in Computer Science from the University of London in 1993, a MSc in 1995 from the University of Leeds and a PhD in Computer Vision from Brunel University in 1999. He is currently a Reader at the University of Surrey, UK where he leads the Cognitive Vision Group within the Centre for Vision Speech and Signal Processing. His research centres on the use of computer vision to locate, track and understand humans. His research into tracking and artificial life received worldwide media coverage, appeared at the British Science Museum and the Minnesota Science Museum. He has won a number of awards including paper prizes for his work on sign language recognition (undertaken as a visiting Research Fellow at the University of Oxford under subcontract from INRIA), as well as the Sullivan Doctoral Thesis Prize in 2000 for the best UK PhD thesis in vision. He was a member of the British Machine Vision Association (BMVA) executive committee and company director for 7 years. He is a London Technology Network Business Fellow, a member of the British Machine Vision Association, a Fellow of the Higher Education Academy and a Senior Member of the Institute of Electrical and Electronic Engineers.

Andrew Gilbert Andrew Gilbert received the BEng (Hons) degree in Electronic Engineering from the University of Surrey, United Kingdom in 2005. He received the PhD degree from Centre for Vision Speech and Signal Processing at the University of Surrey in 2009. His current research interests include real-time visual tracking, human activity recognition, and Intelligent Surveillance. He is currently a Research Fellow at the University of Surrey.