Spatio-temporal Feature Recogntion using Randomised Ferns

Olusegun Oshin, Andrew Gilbert, John Illingworth, Richard Bowden

Spatio-temporal Feature Recogntion using Randomised Ferns

Olusegun Oshin, Andrew Gilbert, John Illingworth, and Richard Bowden

CVSSP, University of Surrey, Guildford, Surrey GU2 7XH United Kingdom
{o.oshin,a.gilbert,j.illingworth,r.bowden}@surrey.ac.uk

Abstract. In this paper we present a generic classifier for detecting spatio-temporal feature points within video. The premise being that given a feature detector, we can learn a classifier that duplicates its functionality which is both accurate and computationally efficient. This means that feature point detection can be achieved independent of the complexity of the original interest point formulation. We extend the naive Bayesian classifier of Ferns to the spatio-temporal domain and learn classifiers that duplicate the functionality of common spatio-temporal interest point detectors. Results demonstrate accurate reproduction of results with a classifier that can be applied exhaustively to video at frame-rate, without optimisation, in a scanning window approach.

1 Introduction

This paper presents an approach to learning a classifier capable of accurately and efficiently detecting spatio-temporal interest points in constant time for use in action recognition. Recognising actions in videos is a popular area in the field of computer vision. The task is complex because detectors need to be able to account for occlusion, cluttered backgrounds, camera motion, scale and illumination changes, and geometric variations in videos. Building upon the success of interest point detection in object recognition, one of the current approaches to the problem involves extracting local spatio-temporal interest points and encoding them using descriptors that are robust to the variations mentioned above. These interest points are local regions that possess high information content. They are usually areas of large local variations in pixel intensity. In 2D images these changes are observed in the spatial dimension. For videos however, temporal variations also have to be considered.

We propose a general framework for training interest point detectors in video, such that, given examples of the type of interest point within a training video, we can detect those feature types at near frame rate in a novel video sequence. We extend a Naive Bayesian classifier called Ferns [1] to the spatio-temporal domain and show that we are able to recognise chosen spatio-temporal features quickly and reliably. Our work is motivated by the use of ferns in rapidly classifying keypoints in 2D images while maintaining high classification rates.

We test our method by comparing its ability to reproduce the functionality of popular spatio-temporal interest point detectors proposed by Laptev and Lindeberg [2] and Dollar *et al* [3]. The key contributions of this paper are the extension

of the Fern classifier to the spatio-temporal domain, hence encoding temporal information of features without increasing computational overhead; and a generic approach to recognising spatio-temporal features efficiently in constant time regardless of original detector complexity.

The layout of the rest of this paper is as follows. Section 2 summarises related work. Section 3 explains how ferns are extended to the spatio-temporal domain. Section 4 presents experimental results from training with two detection algorithms. Finally, section 5 concludes the paper.

2 Related research

An example of a simple spatial interest point detector is the corner detector developed by Harris and Stephens [4], which has been used in many applications involving object matching and tracking. Schmid and Mohr [5] proposed a rotationally invariant local image descriptor at each interest point that has been successful in image indexing. Lowe [6] introduced the SIFT descriptor, which is invariant to scale as well as rotation and translation, but only partially invariant to affine projection and changes in illumination. SIFT has proven effective in many areas as its invariance has underpinned recent trends into large scale object recognition. More recently, considerable effort has gone into the development of affine invariant detectors. An extended evaluation of interest point detectors can be found in [7].

Similarly, in the spatio-temporal domain, interest points are proving to be important in activity recognition. Laptev and Lindeberg [2] extended the Harris Corner detector to include corners in time, where a spatio-temporal corner is essentially a spatial corner exhibiting non-constant motion in time. Dollar *et al* [3] argue that the direct extension of spatial interest point detectors to the spatio-temporal domain does not provide a sufficient method of detecting spatio-temporal features. They found that spatio-temporal features were too sparse so they proposed a method of detecting local features that would maximise discrimination between actions. Other methods based on local features include [8], [9] and [10].

In contrast to the above local detection methods, Ke *et al* [11] and Cooper and Bowden [12] generalise integral images [13] to an efficient space-time representation called an integral volume, allowing for global classification. They use volumetric features which are box volumes reminiscent of the Haar-basis features used by Viola and Jones [13]. Ke *et al* use a sliding volume over the entire video, as an extension of [13] to 3D and build a real time event detector that uses dense optical flow measurements to recognise actions. Other global approaches include [14], [15] and [16].

Of particular relevance is the work of Lepetit *et al* [17], in which they treat wide baseline matching of feature points in images as a classification problem and define a class as the set of all possible views of a particular keypoint. They use Randomised Trees to rapidly match keypoints in images and classify images. Already widely used in character recognition, each node of a randomised tree

contains a simple test which eventually decides the leaf distribution. To increase the speed of randomised trees Ozuysal *et al* [1] proposed the fern classifier. Ferns are non-hierarchical structures that have been shown to achieve excellent classification rates while reducing computational overhead.

3 Method

We aim to detect spatio-temporal features accurately and efficiently in video using a semi-naive Bayesian classifier called Ferns, proposed in [1], as an alternative to Randomised Trees [17]. We learn ferns on regions surrounding spatio-temporal features, which are detected using a spatio-temporal feature detector. In this section we give an overview of the methods used and how they are combined for the training and classification of actions in new videos.

3.1 Randomised Trees and Ferns

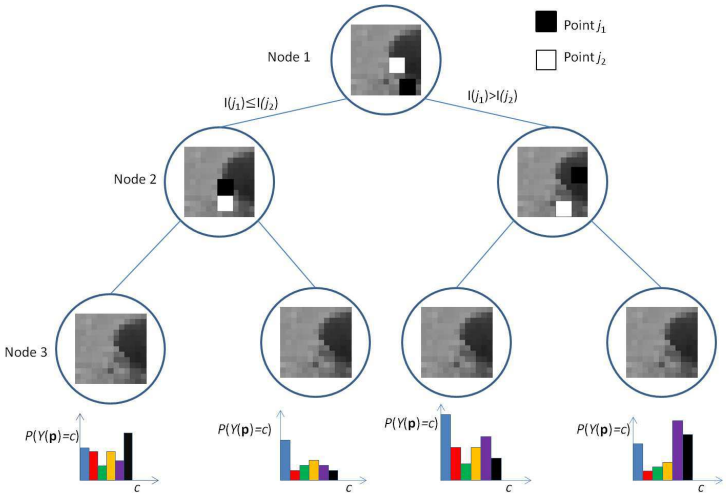

Fig. 1. A Randomised Tree using pixel intensity comparisons at random points in the image patch. After training, each leaf node contains class probabilities of a new patch arriving at that node.

Randomised Trees are simple yet effective classification structures that have been used in handwriting recognition [18] and applied to object recognition [17]. All non-leaf nodes of a randomised tree contain a simple test that splits the space of the data to be classified. Lepetit and Fua [17] used the simple test of comparing pixel intensities, which could be chosen randomly or by a greedy algorithm that gives the best separation based on information gain. After training, the leaf

nodes contain estimates of the posterior distributions over the classes that an image reaches that leaf. Randomised trees are built in a top-down manner and all training examples are passed down the tree through nodes based on the test. Figure 1 shows a randomised tree that uses a comparison of pixel intensities as the test. If the intensity at point j_1 of the image patch is greater than that of point j_2, then the patch is passed down the tree via the right child node; otherwise, it traverses the left child node. The process terminates when the node receives too few examples or it reaches a given depth. It would be very difficult to build one tree that performs well for a large number of classes, training examples and possible tests, so multiple trees are grown, and each tree is trained with a random subset of the training examples to obtain weak dependency between the trees. When classifying, a test patch is passed down each of the trees to a leaf node. The class assigned to the image is found by averaging the posterior distributions at the node of each tree where the image terminates, and the class with the maximum value is selected.

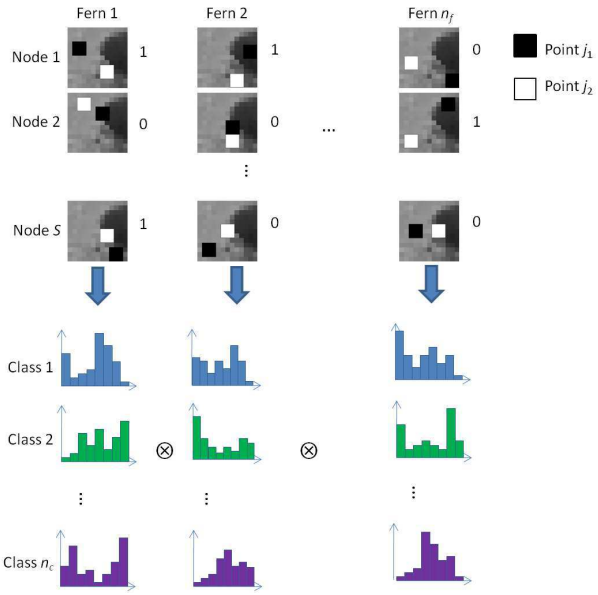

Fig. 2. A set of ferns. The binary digits are results of the comparison between pixels at points j_1 and j_2. During training, an image patch belonging to a particular class is run through all nodes in all ferns and each fern updates its distribution for that class based on the results. During classification, the results are used to select bins in the class histograms.

Ferns [1], unlike randomised trees, are non-hierarchical structures. Each fern consists of a set of ordered binary tests and returns the probability that a patch belongs to each of the classes learnt during training. The structure of ferns is

shown in Figure 2. Each node in a fern returns a binary digit based on the result of the test. Hence, a fern with S nodes will return a number between 0 and $2^S - 1$. For multiple patches that belong to the same class, the output of a fern for that class can be modelled with a multinomial distribution. During classification, performing the set of tests on a test patch also returns a binary code, which is used to obtain the likelihood of occurrence in each class. The class is found by multiplying class probabilities across the ferns, and selecting the class that gives the maximum product. Performance-memory trade-offs can be made by changing the size and number of ferns allowing for flexible implementation.

3.2 Semi-Naive Bayesian Spatio-temporal Classification

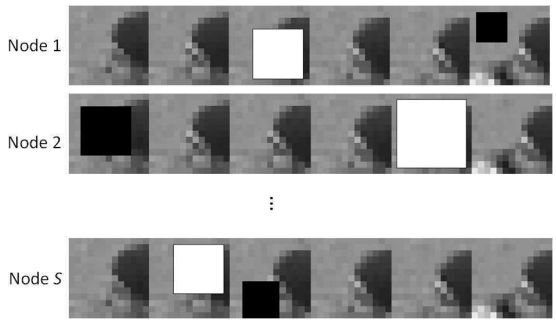

Fig. 3. Example of nodes in the spatio-temporal domain. The cuboid is flattened with respect to time to show subpatches at different temporal offsets.

We follow the method of Ferns described above and extend it to spatio-temporal neighbourhoods centered at feature points called cuboids [3]. The sets of binary tests we perform on the cuboids are not single pixel comparisons. Instead, we have defined a node as a comparison between the sums of two normalised *subpatches*. These are random sized patches that fit within the spatial area of the cuboid. The locations in space and time are also chosen randomly. Figure 3 shows a flattened spatio-temporal volume with subpatches generated on different frames within the volume. Hence for subpatches encoded by a spatial offset x, y, spatial extent x_s, y_s and temporal offset t, a node is a simple comparison:

$$f_j = \begin{cases} 1 \text{ if } s(x_1, y_1, x_{s_1}, y_{s_1}, t_1) < s(x_2, y_2, x_{s_2}, y_{s_2}, t_2); \\ 0 \text{ otherwise.} \end{cases}$$

where s is the sum of a subpatch, given by

$$s(x, y, x_s, y_s, t) = \frac{1}{x_s y_s} \sum_{x'=x}^{x+x_s} \sum_{y'=y}^{y+y_s} I(x', y', t).$$

The sums are normalised by the number of pixels in the subpatch, $x_s y_s$. We make use of sum of subpatch comparisons as opposed to pixel comparisons so as to build a classifier that is more robust to scale variations.

3.3 Training

Given a cuboid belonging to a particular class, we train a fern classifier by generating S nodes as described above. We perform the node tests and obtain values f_j, where $j = \{1 \dots S\}$, for each fern. All the node results $\{f_1, f_2, \dots, f_S\}$ are combined to form a binary code, which is then converted to decimal. For a set of cuboids belonging to the same class, we build a histogram of the decimal values, and normalise it to give a probability distribution for that class. For additional classes, the same tests are performed within the fern and separate distributions are obtained for them. Multiple ferns are trained by generating new nodes and obtaining distributions for all classes within the ferns.

In order to rapidly calculate the difference between the sums of subpatches, we calculate the integral image representation [13] of each frame in the cuboid. We choose to use integral images of each frame as opposed to an integral volume of the cuboid because the volumetric approach adds an additional parameter. Laptev and Perez [19] investigated the use of a classifier trained on a single keyframe of drinking actions. They bootstrap a space-time classifier and apply it to the keyframes, and have shown that this method of combining shape and motion information gives significant improvement in detection performance.

3.4 Classification

To classify a new cuboid, we apply the fern classifier by performing the same set of ordered node tests on the cuboid as we did during training, and obtain a decimal value from the combined binary node results. The decimal value determines the bin of the class distributions within the fern that is selected. For a single fern, the cuboid most likely belongs to the class that has the highest value in the selected bin. For multiple ferns, we combine the selected bin values within the classes and across the ferns, assuming independence between the distributions. We select the class that has the maximum likelihood.

4 Results

We train and test our algorithm on the KTH video data set [8], which contains videos of 25 persons performing six actions in four different scenarios. These actions are boxing, hand waving, walking, running, jogging and hand clapping. The videos are taken over static homogenous backgrounds and are split into four different conditions that include scale variations, different clothes and lighting changes. Of the 25, 8 persons are used for training, 9 for testing, and the remainder for validation purposes as outlined in the data set. Although the KTH Data set is limited in terms of complexity, it is important to note that our intention

is to provide comparable interest point segmentation to other detectors and not to classify the actions.

We evaluate against two different feature point detectors and try to obtain comparable detection performance. Having trained the classifier, testing is done using each one of the test videos. We present our detector's ability to recognise learnt features in the video using a scanning volume over the entire test video sequence. The feature point detectors used are described in the next section. The first is an extension of the Harris corner detector [4] to the spatio-temporal domain by Laptev and Lindeberg [2], selecting local maxima of spatio-temporal corners as features. In our results this method is labelled *Laptev*. The second method by Dollar *et al* [3] applies separable linear filters to the video and selects local maxima of the response as features, labelled *Dollar*.

4.1 Feature Point Extraction

Fig. 4. Example frames from action video sequences highlighting points where spatio-temporal features have been detected. These points are used as positive data during training.

Laptev and Lindeberg [2] extend the Harris corner detector [4] to the spatio-temporal domain by requiring that image values in space-time have significant variations in the spatial and temporal dimensions. Hence these spatio-temporal corners are spatial corners exhibiting non-constant motion in a spatio-temporal neighbourhood. They compute a windowed 3×3 second moment matrix composed of first order spatial and temporal derivatives, averaged with a Gaussian weighting function. Interest points are then detected by searching for regions that have significant eigenvalues of the matrix.

The second method, proposed by Dollar [3], applies separable linear filters to the video sequence. A response function of the form $R = (I * g * h_{ev})^2 + (I * g * h_{od})^2$ is obtained, 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 quadrature pair of 1D Gabor filters applied in the temporal dimension. The detector responds best to complex motions made by regions that are distinguishable spatially, including spatio-temporal corners as defined by [2], but not to pure translational motion

or motions involving areas that are not distinct in space. Local maxima of the response function R are selected as interest points, and *cuboids* are extracted, which are the windowed pixel values around the interest point in the spatial and temporal dimensions.

For both methods, the scale at which an event is observed determines the type of event that is detected, so interest points are computed at different spatio-temporal scales.

4.2 Performance

During training, we apply the classifier to the extracted volumes surrounding each of the feature points detected. Figure 4 highlights detected feature points in a video. We detect features at one spatial and temporal scale. Ideally, training should be done at different spatio-temporal scales and the scanning volume applied over a range of scales, but we show that our classifier is able to handle scale changes. Once feature volumes are extracted, we artificially generate additional positive example cuboids by randomly offsetting and scaling the cuboids at the detected feature points by a small amount in all directions, making our classifier robust to slight variations in the feature regions. We then create a class of negative example cuboids, which are extracted at randomly selected areas that do not overlap with feature cuboids. The number of background cuboids and the number of feature cuboids are equal to avoid biasing the classifier. The number of ferns and the number of nodes in each fern are parameters that can be varied to achieve a trade-off between memory and performance. For our tests we generate 50 ferns with 3 nodes.

In order to detect features in a new video we make use of a spatio-temporal scanning window across the video. We then apply the ferns classifier on the region within the window to determine whether it is a feature or background. The subpatch comparisons are done efficiently by using an integral image representation [13] of the entire video, which is calculated once. This differs from the integral video used in [11] as the integral images are computed on each frame of the video separately. The sums of patches selected can therefore be computed using 4 array dereferences, giving 8 array dereferences for each node comparison. The scanning window operates over alternate pixels and frames to reduce classification time without loss of accuracy (the random noise added during training which gives robustness to offsets from the optimal detection position).

To determine whether the windowed volume contains a feature, we find the log likelihood ratio that the cuboid contains a feature to background. Receiver Operating Characteristic (ROC) Curves are used to determine how well the classifier is able to emulate the detections learnt during training. Positives in our data are feature cuboids detected by the feature detector prior to training, and true positives are positives that have also been labelled a feature by our classifier. The ROC curve is obtained by varying the threshold on the likelihood ratio.

We expect our classifier to trigger multiple detections around a positive and that the detections will not always occur at the same points selected by the

feature detectors. This is natural as noise was added during training. To this end, we choose a minimum percentage by which a cuboid classified as a feature must overlap an actual detected feature before it can be labelled a true positive. Figure 5 shows averaged ROC curves obtained for different percent overlaps. All percent overlaps perform well, with 100% overlap having a surprisingly good Equal Error Rate of approximately 0.65. For our experiments we deemed 50% overlap fair.

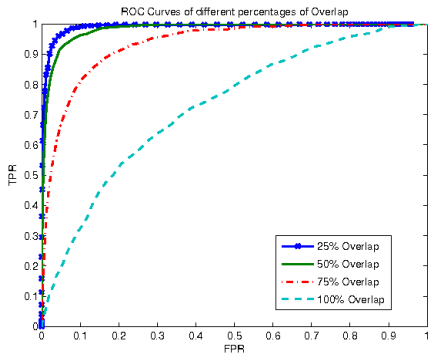

Fig. 5. Classification of features against background for various percentages of overlap between classified feature cuboids and actual positive feature cuboids

Figure 6 shows how well we are able to detect learnt features in a novel video sequences. Each curve indicates the different scenarios of the KTH video data set [8], where $S1$ is static with a homogenous background, $S2$ includes scale variations, $S3$ is performed with different clothing, and $S4$ has illumination variations. We display results for Boxing, Handwaving and Walking for both feature point detectors. It can be seen that the spatio-temporal fern classifier performs extremely well in detecting learnt features. It is able to correctly recognise more than 95% of the *Dollar* features with less than 5% error for all the scenarios. For *Laptev* features, we obtain Equal Error Rates of approximately 90% for the actions in all scenarios except for Illumination variations ($S4$), where the Equal Error Rates are roughly 85% for Boxing and Walking features. This lower performance on the *Laptev* features is not surprising as these features are more localised than those of *Dollar* and the spatial-temporal volume the ferns operate upon incorporates additional surrounding information. We kept the classifier the same between both experiments to allow comparison, but optimising the spatio-temporal support region of the classifier for different features is likely to increase performance further. It is worth noting that although the classifier was trained at one spatial and temporal scale, the $S2$ curves show that performance on different scales is comparable to that of the static homogenous background. This highlights the classifier's ability to cope with changes in scale from the training data. To further assess generality, we investigated the ability of our classifier to

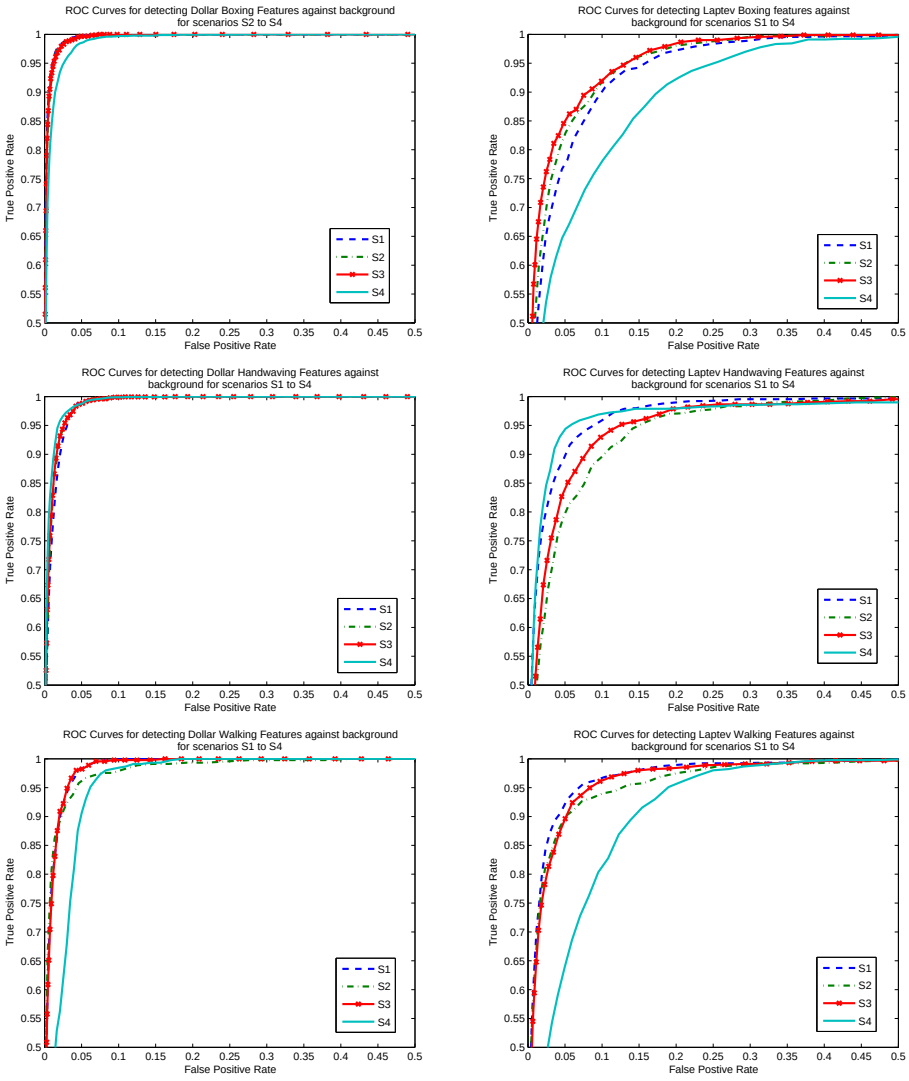

Fig. 6. Left: ROC Curves for features detected by the *Dollar* feature detector against background for Boxing, Handwaving and Walking. Right: ROC Curves for features detected by the *Laptev* feature detector against background for Boxing, handwaving and Walking. Each graph shows performance in the four scenarios of the KTH video data set, S1 - S4, where S1 videos have a static homogenous background, S2 have scale variations, subjects in S3 have different clothes, and S4 have illumination variations.

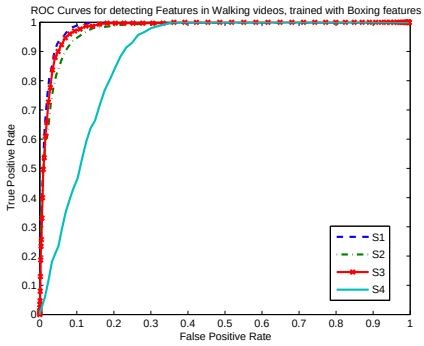

Fig. 7. ROC curves for features detected in Walking video sequences trained with Boxing features. S1 to S4 are defined as in Figure 6

detect from one action, features learnt from other actions. This test relies upon the original features being invariant to action type as well occurring in both activities, which although desirable, might be assumed is not the case. Having trained the classifier on Boxing features, we test its ability to detect features that occur in a Walking video sequence and present the results in Figure 7. It can be seen that features common to Boxing and Walking are detected with high equal error rates, showing that irrespective of the action, the classifier gives good performance in detecting spatio-temporal features. This also goes some way to demonstrating that the features are indeed consistent and invariant to the action performed.

The Dollar detector runs at frame rate on the KTH data which is impressive despite its matlab implementation. The Laptev detector is considerably slower to apply, but again, this is a matlab implementation and could undoubtedly be sped up through optimisation. Our classifier was implemented in C++ and using the scanning window approach also achieves frame rate processing on the KTH data. However, the speed of the classifier is dependent upon the number of Ferns and nodes used for classification. With optimisation, similar classification performance could be achieved with less ferns and future work will investigate this optimisation. Furthermore, Ferns can be structured so that a low cumulative likelihood early in classification is rejected without applying the entire classifier.

5 Conclusion

We extended the Semi-Naive Bayesian classifier to the spatio-temporal domain and proposed a generic approach to recognising spatio-temporal features. The method is demonstrated on the task of detecting learnt spatio-temporal features on the KTH video data set. We presented results of detection having trained on features selected by two different spatio-temporal feature point detectors and show the ability of the classifier to obtain results comparable to these feature

detectors on test data. These high classification rates are achieved in constant time without an increase in computational overhead.

References

1. Ozuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2007) 1–8
2. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proc. IEEE Intl. Conf. on Computer Vision. Volume 2. (2003) 432–439
3. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Proc. IEEE Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. (2005) 65–72
4. Harris, C., Stephens, M.: A combined corner and edge detection. In: Proc. Alvey Vision Conf. (1988) 147–151
5. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 530–534
6. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. IEEE Intl. Conf. on Computer Vision. (1999) 1150–1157
7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1615–1630
8. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: Proc. Intl. Conf. on Pattern Recognition. Volume 3. (2004) 32–36
9. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. Intl. Journal of Computer Vision **79** (2008) 299–318
10. Oikonomopoulos, A., Patras, I., Pantic, M.: Spatiotemporal salient points for visual recognition of human actions. IEEE Transactions on Systems, Man, and Cybernetics, Part B **36** (2006) 710–719
11. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: Proc. IEEE Intl. Conf. on Computer Vision. Volume 1. (2005) 166–173
12. Cooper, H., Bowden, R.: Sign language recognition using boosted volumetric features. In: Proc. IAPR Workshop on Machine Vision and Application. (2007) 359–362
13. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2001) 511–518
14. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: Proc. Intl. Conf. on Computer Vision, 2003. (2003) 726–733
15. Shechtman, E., Irani, M.: Space-time behavior-based correlation. IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007) 2045–2056
16. Wong, S.F., Cipolla, R.: Extracting spatiotemporal interest points using global information. In: Proc. IEEE Intl. Conf. on Computer Vision. (2007) 1–8
17. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 1465–1479
18. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation **9** (1997) 1545–1588
19. Laptev, I., Perez, P.: Retrieving actions in movies. In: Proc. IEEE Intl. Conf. on Computer Vision. (2007) 1–8