Complex activity recognition and anomaly detection in multimedia streams

By Ioannis Kaloskampis∗, Yulia A. Hicks and David Marshall
Cardiff University, Queens Buildings, The Parade, Cardiff CF24 3AA, Wales, UK.

Abstract

This article presents a framework for activity recognition and anomaly detection in multimedia streams featuring complex human activities. The framework models human activities as temporal sequences of their constituent actions and can handle actions that occur concurrently in multiple parallel streams. It operates in a supervised manner and comprises three stages, which are extraction of action sequences from data streams, feature selection and activity recognition/anomaly detection.

The framework is assessed on the 'bridge design' dataset which is based on a real-life application. Preliminary results show that the proposed framework, when used in conjunction with standard classifiers offers good classification accuracy in activity recognition and anomaly detection.

1. Introduction

Modelling human activities as temporal sequences of their constituent actions has been the object of much research effort in recent years. Most of this work concentrates on tasks where the action vocabulary is relatively small and/or each activity can be performed in a limited number of ways. In this article, we propose a robust framework for recognising prolonged activities and detecting anomalies in tasks which can be effectively achieved in a variety of ways.

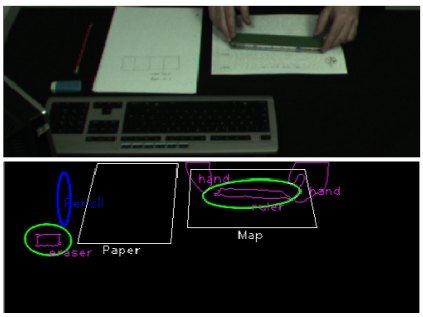
There are currently four approaches to activity modelling: (1) grammar-driven representations, *e.g.* (Ivanov & Bobick (2000)); (2) vector space models (VSMs), *e.g.* (Stauffer & Grimson (2000)), where an activity is represented as a vector of its constituent actions; (3) local event statistic methods, which capture neighbouring temporal relations between an activity's constituent actions; (4) statistical graphical models, such as the hidden Markov model (HMM) (Rabiner (1989)) and its extensions.

Most methods assume that actions constituting activities take place in a nonparallel manner. Additionally, a single data stream is typically used.

The contributions of this article are: (1) a framework for activity recognition and anomaly detection of complex activities in multimedia streams is proposed; contrary to existing methods, the framework is capable of analysing actions which can occur concurrently in multiple parallel streams, (2) the 'bridge design' dataset is introduced, which is based on a real-life application.

The paper is structured as follows. Section 2 describes the method used to extract action sequences from data streams; the feature selection process is covered in section 3. Section 4 explains how activities are recognised and anomalies are detected. Experimental results are presented in section 5; the paper is concluded in section 6.

∗ This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1 and the MOD University Defence Research Collaboration in Signal Processing.



(a)

Class 1: Erasing	[(hand 1)T(eraser) | | (hand 1)T(eraser)]&& (eraser)T(paper) [(hand 1)S(eraser) | | (hand 2)S(eraser)] && (eraser)S(paper) [(hand 1)S(eraser) | | (hand 2)S(eraser)] && (eraser)T(paper) [(hand 1)T(eraser) | | (hand 2)T(eraser)] && (eraser)S(paper)
Class 2: Writing/ Sketching/ Waiting	[(hand 1)T(pencil) | | (hand 2)T(pencil)] && (pencil)T(paper) [(hand 1)S(pencil) | | (hand 2)S(pencil)] && (pencil)S(paper) [(hand 1)S(pencil) | | (hand 2)S(pencil)] && (pencil)T(paper) [(hand 1)T(pencil) | | (hand 2)T(pencil)] && (pencil)S(paper)
Class 3: Measuring on map	[(hand 1)T(ruler) | | (hand 2)T(ruler)] && (ruler)T(map) [(hand 1)S(ruler) | | (hand 2)S(ruler)] && (ruler)S(map) [(hand 1)S(ruler) | | (hand 2)S(ruler)] && (ruler)T(map) [(hand 1)T(ruler) | | (hand 2)T(ruler)] && (ruler)S(map) (hand 1)T(ruler) && (ruler)T(map) && (hand 2)T(ruler) (hand 1)T(ruler) && (ruler)T(map) && (hand 2)S(ruler) (hand 1)S(ruler) && (ruler)S(map) && (hand 2)T(ruler) (hand 1)S(ruler) && (ruler)S(map) && (hand 2)S(ruler) (hand 1)S(ruler) && (ruler)T(map) && (hand 2)T(ruler) (hand 1)S(ruler) && (ruler)T(map) && (hand 2)S(ruler) (hand 1)T(ruler) && (ruler)S(map) && (hand 2)T(ruler) (hand 1)T(ruler) && (ruler)S(map) && (hand 2)S(ruler)

(b)

Figure 1: **(a):** Detection of action *measuring*. **(b):** Mapping from QSR to actions for 3 action classes. Subclasses of class 2 are distinguished by analysing the moving hand's trajectory. Symbols used: T : Touches, S : Surrounds, && : And, $\|$ Or.

2. Extracting action sequences

Our sequence classification algorithm analyses time series of actions representing complex human activities. In this section we present a method to extract these actions from video of a human interacting with various objects on a table.

In our study scene all important objects are at least partially visible at all times. To monitor their movements, one video tracker is placed on each object at the first frame of each sequence. Tracking is performed using a colour histogram-based observation model and a second order autoregressive dynamical model as in Pérez *et al.* (2002). Each object in the scene is now represented by a tracking window.

We extract actions from video footage by identifying patterns of *qualitative spatial relations* (QSR) (Sridhar *et al.* (2008)) between the tracking windows of moving objects. Two such windows representing objects can be (1) spatially Disconnected *(D)*, (2) connected through the surrounds *(S)* relationship, (3) connected through the Touches *(T)* relationship. An action p_x, starts at time point t_i when the spatial relations defining it start co-occurring and ends at time point t_j when the relations stop existing. With this method, simple actions like measuring are detected (Fig. 1a). The set of possible object interactions is specified *a priori*; *e.g.*, spatial relationship {Hand (T)ouches Ruler} and {Map (S)urrounds Ruler} is interpreted as the action *measuring*. The mapping from QSR to actions for three action classes is shown in Fig. 1b.

Certain actions can be qualitatively similar, *e.g. writing*, *sketching* and *waiting*. By statistically analysing the motion trajectories of the objects involved in these actions we can disambiguate between these actions. This analysis is performed with a continuous HMM. In our experiments, the model was trained with 750 sequences, each of length 15 seconds, with 250 sequences representing each class (*writing*, *sketching* and *waiting*).

Extraction of actions from input streams with the aid of QSR framework offers a basic understanding of the studied task. The resulting data is in form:

$$Q = \{p_a(t_{a,s}), p_b(t_{b,s}), p_b(t_{b,e}), p_a(t_{a,e}), \dots\} \tag{2.1}$$

with $p_x(t_{x,s}), p_x(t_{x,e})$ start and end of a sub-activity or action p_x. Note that this representation can handle multiple parallel streams. This can be achieved by (1) extracting a sequence in the form of *Eqn.*2.1 for each stream, (2) concatenating all extracted sequences into a super-sequence Q_{total}, (3) placing all elements of Q_{total} in chronological order. Two streams were used in this work, one resulting from video as described in section 2 and another one which is used to record actions not observable in the video stream.

A brief description of the second stream is given in the Appendix; for more details please see Kaloskampis *et al.* (2011).

3. Feature selection

When the sequences representing activities only contain actions directly related to the activity performed, encoding temporal relations between actions and determining discriminative features is an easy task for modern classification algorithms. However, actions which are irrelevant to the performed activity are sometimes encountered in action sequences. Such actions in general make the classification task more difficult. In this work two feature selection approaches which can detect irrelevant actions are investigated, specifically RF variable importance (RFVI) (Breiman (2001)) and SVM variable importance (SVMVI) (Maldonado & Weber (2009)).

4. Activity recognition and anomaly detection

The action sequences are identified using a supervised classifier which operates in two phases, a training phase and an identification phase. During the training phase, the classifier is built automatically using data labelled by experts. In the identification phase, novel data (*i.e.* data not used during the training phase) are fed to the classifier which assigns them to classes. Each of these classes represents a complex activity.

Anomaly detection is handled as follows. When building the classifier, two classes correspond to each activity: one for correct and another one for erroneous executions of the activity. Anomalies are the sequences which are assigned by the classifier to classes corresponding to erroneous activity executions.

Three widely used activity recognition classifiers are tested in this article, which are random forests (RF) (Breiman (2001)), HMMs and support vector machines (SVM) (Cortes & Vapnik (1995)).

5. Experimental results

The framework is assessed on the 'bridge design' dataset which is based on a real-life application. The dataset is available by contacting the first author of this article. Six civil engineering professionals and 14 civil engineering students were asked to solve a bridge design task and were recorded while working on it. The recordings comprise two streams: (1) video footage of the engineers's interactions with objects on a table (2) a stream listing their interactions with specialised software. From the recordings 54 sequences were extracted (each of length 5-15 minutes) to serve as the training set. In each sequence, participants execute one of three complex tasks: evaluate soil condition, estimate transient loads and evaluate bridge cost. The test data is a different set of 72 sequences (36 correct and 36 erroneous executions) obtained in a similar way.

Preliminary results on this dataset using RF, HMMs and SVM classifiers are reported here. Each classifier was applied with and without feature selection. For the RF classifier the average classification accuracy over 10 runs is reported.

Only the training dataset was used to select the important features. Both tested algorithms, RFVI and SVMVI, output the importance for all features. The process was repeated 10 times and the average importance for all features was estimated. In the case of RFVI features with negative importance were considered redundant and are removed. For SMVI no suitable threshold to filter out unimportant features was found.

The results are shown on Table 1. The application of RFVI algorithm results in an overall increase of the classification accuracy for all tested algorithms. This increase is significant for the HMM classifier, as activity recognition improves by 6% and marginal for RF, where the anomaly detection rate increases by 3%. In the case of SVM, anomaly

	HMM		**SVM**		**RF**	
	Corr.	Anom.	Corr.	Anom.	Corr.	Anom.
No feature selection	72	61	86	58	81	78
SVM Variable Importance	72	61	86	58	81	78
RF Variable Importance	78	61	83	64	81	81

Table 1: Classification results in terms of percent classification accuracy for three classifiers and two feature selection algorithms.

detection rate increases by 6% and recognition rate decreases by 3%. The SVMVI method does not offer any performance gain as it does not filter out redundant features.

6. Conclusion

We have presented a framework for activity recognition and anomaly detection of complex activities in multimedia streams. The framework models human activities as temporal sequences of their constituent actions. It is capable of handling actions that can occur concurrently in multiple parallel streams. Additionally, the 'bridge design' dataset was introduced, which is based on a real-life application. Preliminary results were presented on this dataset, using widely used classifiers and feature selection methods in conjunction with the proposed framework. The classification and feature selection tasks will be further investigated in future work.

A. Cognitive action detection

In the bridge design task, actions such as *choose bridge type* and *estimate soil condition* occur which cannot be identified through the interaction of the participant with scene objects. We define such actions as *cognitive actions*. To detect them, we monitor the user's interactions with a knowledge based system (KBS). Each interaction with the KBS is linked to a specific cognitive action. Therefore, when the user interacts with the KBS, we can deduce which cognitive action is performed. The KBS records the time at which an interaction occurs so that the cognitive action can be placed within the activity time line which is given by *Eqn.*2.1.

REFERENCES

Ivanov, Y. & Bobick, A. 2000 Recognition of visual activities and interactions by stochastic parsing. *PAMI* **22**(8):852–872.

Stauffer, C. & Grimson, W.E. 2000 Learning patterns of activity using real-time tracking. *PAMI* **22**(8):747–757.

Rabiner, L. R. 1989 A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77**(2), 257–286.

Breiman, L. 2001 Random Forests. *Machine Learning* **45**(1), 5–32.

Cortes, C. & Vapnik, V. 2001 Support Vector Networks. *Machine Learning* **20**, 273–297.

Maldonado, S. & Weber, R. 2009 A wrapper method for feature selection using Support Vector Machines. *Information Sciences* **179**(13), 2208–2217.

Sridhar, M., Cohn, A.G., & Hogg, D.C. 2008 Learning Functional Object-Categories from a Relational Spatio-Temporal Representation. *Frontiers in Artificial Intelligence and Applications* **178**, 606–610.

Pérez, P., Hue, C., Vermaak J. & Gangnet, M. 2002 Color-based probabilistic tracking. In *Proceedings of ECCV* **2**, 661–675.

Kaloskampis, I., Hicks, Y.A., & Marshall, D. 2011 Reinforcing conceptual engineering design with a hybrid computer vision, machine learning and KBS framework. In *Proceedings of SMC*, 3242–3249.