Chapter 1

A Logic-Based Approach to Activity Recognition

Alexander Artikis
NCSR Demokritos, Greece

Marek Sergot
Imperial College, UK

Georgios Paliouras
NCSR Demokritos, Greece

ABSTRACT

The authors have been developing a system for recognising human activities given a symbolic representation of video content. The input of the system is a stream of time-stamped short-term activities detected on video frames. The output of the system is a set of recognised long-term activities, which are pre-defined spatio-temporal combinations of short-term activities. The constraints on the short-term activities that, if satisfied, lead to the recognition of a long-term activity, are expressed using a dialect of the Event Calculus. The authors illustrate the expressiveness of the dialect by showing the representation of several typical complex activities. Furthermore, they present a detailed evaluation of the system through experimentation on a benchmark dataset of surveillance videos.

1. INTRODUCTION

A common approach to human activity recognition separates low-level from high-level recognition. The output of the former type of recognition is a set of activities taking place in a short period of time: ‘short-term activities’. The output of the latter type of recognition is a set of ‘long-term activities’, that is, pre-defined spatio-temporal combinations of short-term activities. We focus on high-level recognition.

We define the set of long-term activities of interest, such as ‘fighting’ and ‘meeting’, as combinations of short-term activities - for example, ‘walking’, ‘running’, and ‘inactive’ (standing still) - using a logic programming implementation of the Event Calculus (Kowalski & Sergot, 1986; Artikis & Sergot, 2010). More precisely, we
employ the Event Calculus to express the temporal (and other) constraints on a set of short-term activities that, if satisfied, lead to the recognition of a long-term activity.

In this chapter we extend our previous work on activity recognition (Artikis & Paliouras, 2009) in the following ways. First, we use a more efficient Event Calculus dialect and implementation to compute the intervals of long-term activities. Second, we illustrate the expressiveness of the proposed Event Calculus dialect by presenting several complex activity definitions. We are able to construct much more succinct representations of activity definitions for video surveillance than we had in our earlier work. Third, we present a more detailed and informative evaluation of the Event Calculus on activity recognition. We show through experimentation how incomplete short-term activity narratives, inconsistent annotation of short-term and long-term activities, and a limited dictionary of short-term activities and context variables affect recognition accuracy. Fourth, we evaluate our approach on a dataset with a refined dictionary of short-term activities, in order to validate experimentally our intuition that a finer classification of short-term activities increases, under certain circumstances, the accuracy of long-term activity recognition. Indeed, the refined dictionary of short-term activities—which can be provided by state-of-the-art short-term activity recognition systems—together with the updated long-term activity definitions presented in this chapter, lead to much higher Precision and Recall rates.

The remainder of the chapter is organised as follows. First, we present the Event Calculus dialect that we employ to formalise activity definitions. Second, we describe the dataset of short-term activities on which we perform long-term activity recognition. Third, we present our knowledge base of long-term activity definitions. Fourth, we present our experimental results. Finally, we discuss related work and outline directions for further research.

### 2. THE EVENT CALCULUS

Our Long-Term Activity Recognition (LTAR) system consists of a logic programming (Prolog) implementation of an Event Calculus dialect. The Event Calculus, introduced by Kowalski and Sergot (1986), is a many-sorted, first-order predicate calculus for representing and reasoning about events and their effects. For the dialect used here, hereafter LTAR-EC (event calculus for long-term activity recognition), the time model is linear and it may include real numbers or integers. Where \( F = V \) denotes that fluent \( F \) has value \( V \). Boolean fluents are a special case in which the possible values are true and false. Informally, \( F = V \) holds at a particular time-point if \( F = V \) has been initiated by an event at some earlier time-point, and not terminated by another event in the meantime.

An event description in LTAR-EC includes axioms that define, among other things, the event occurrences (with the use of the happensAt and happensFor predicates), the effects of events (with the use of the initiatedAt and terminatedAt predicates), and the values of the fluents (with the use of the initially, holdsAt and holdsFor predicates).

Table 1 summarises the main predicates of LTAR-EC. Variables, starting with an upper-case letter, are assumed to be universally quantified unless otherwise indicated. Predicates, function symbols and constants start with a lower-case letter.

The domain-independent axioms for holdsAt and holdsFor are such that, for any fluent \( F \), holdsAt(\( F = V, T \)) if and only if time-point \( T \) belongs to one of the maximal intervals of \( I \) such that holdsFor(\( F = V, I \)). However, for efficiency the implementation employs different procedures for these two tasks, and various indexing techniques to reduce search and improve efficiency further. Briefly, to compute holdsFor(\( F = V, I \)), we find all time-points \( Ti \) in which \( F = V \) is initiated, and then, for each \( Ti \), we compute the first time-point after \( Ti \) in which \( F = V \) is terminated.