Activity Recognition in Wide Aerial Video Surveillance Using Entity Relationship Models

Jongmoo Choi, Yann Dumortier, Jan Prokaj, and Gérard Medioni
Computer Vision Lab, Institute for Robotics and Intelligent Systems
University of Southern California, USA
{jongmooc,prokaj,medioni}@usc.edu,yann.dumortier@gmail.com

ABSTRACT

We present the design and implementation of an activity recognition system in wide area aerial video surveillance using Entity Relationship Models (ERM). In this approach, finding an activity is equivalent to sending a query to a Relational DataBase Management System (RDBMS). By incorporating reference imagery and Geographic Information System (GIS) data, tracked objects can be associated with physical meanings, and several high levels of reasoning, such as traffic patterns or abnormal activity detection, can be performed. We demonstrate that different types of activities, with hierarchical structure, multiple actors, and context information, are effectively and efficiently defined and inferred using the ERM framework. We also show how visual tracks can be better interpreted as activities by using geo information. Experimental results on both real visual tracks and GPS traces validate our approach.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - *Spatial databases and GIS*

General Terms

Algorithms, Experimentation

Keywords

Wide aerial surveillance, activity recognition

1. INTRODUCTION

Our goal is to provide an efficient activity recognition framework for wide area aerial video surveillance where vehicular segmented tracks are the essential components. The input to our activity recognition framework consists of geo-registered tracks inferred by a tracking module. Activities are defined as tracks associated with certain properties and their relationships with one or more objects (be it tracks, or other georeferenced entities). Since an activity may involve a sequence of motion patterns (events) and multiple actors, how to represent events and activities is a challenging task.

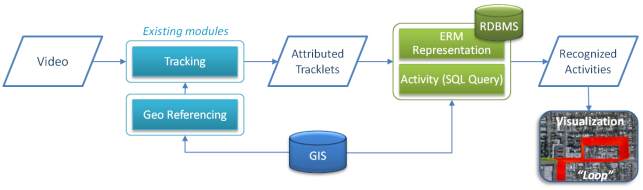

Figure 1: Overview of the proposed approach.

We propose to define and recognize a large number of activities with the Entity Relationship Model (ERM) [3] framework (Fig. 1). The ERM is an appropriate framework to capture multiple relationships between elements, which allows us to efficiently represent hierarchical structures, multiple actor activities, and context information. We use a RDBMS (Relational DataBase Management System) [1], to store and retrieve all meta-data in our activity recognition system, including tracking results, geospatial objects and context information, and use Structured Query Language (SQL) [1] to define and recognize activities. In this approach, finding an activity is equivalent to sending a set of SQL statements to the RDBMS. As an additional benefit, RDBMS scales well to a distributed system to handle large amounts of data.

2. RELATED WORK

We briefly review the literature on activity recognition in wide area surveillance. Reilly et al. [11] shows object detection and tracking in a wide area surveillance domain, where bipartite graph matching and linking tracks were applied to detection results, and grid cells were employed to provide a set of local scene constraints such as road orientation and object context for tracking. Pollard et al. [8] presented activity detection results using a complex probabilistic framework but only a single activity, convoys, was presented and geospatial constraints were not considered. In [5], high-level complex event inference from multimodal data using Markov Logic Networks is presented for wide area surveillance. A framework for semi-supervised nonlinear embedding methods, based on a neural network optimizing the graph-based cost function, to analysis large-scale spatio-temporal network data is presented [9]. In [6], large sets of mobile objects' trajectories are distributed to a network of database servers

ACM SIGSPATIAL '12, November 6-9, 2012. Redondo Beach, CA, USA

Table 1: ERM representation

Entity	
	track point, tracklet, track, traffic rule,
	road segment, building, area, $\cdots$
Relationship	
	building -belong to- road segment
	tracklet -is on- road segment
	road segment -has- traffic rule
	must_stop -is a- traffic rule, $\cdots$
Event can be represented by a relationship	
	tracklet (track_id, $\cdots$, speed=95)
	tracklet (track_id, $\cdots$, road_id)
	road (road_id, $\cdots$, speed_limit)
	speeding: tracklet.speed > road.speed_limit

by using Space-partitioned Moving Objects Databases (SPMODs).

Given our domain, where vehicular segmented tracklets are the essential components, we show that activities can be effectively and efficiently inferred using a relational database model.

3. ERM-BASED ACTIVITY RECOGNITION

3.1 Computing tracklets from imagery

The atomic spatio-temporal information in our system is called a tracklet, a segmented portion of a track representing vehicle's "instantaneous" motion, such as going straight, turning left or turning right. Each tracklet has a collection of attributes $x_i = \{\lambda_1, \lambda_2, \cdots, \lambda_m\}$, where an element λ_i presents a physical property such as time, location, and speed.

We use a state of the art real-time tracker [10], which has been used by Lawrence Livermore National Lab. Since the most important parts of the track are those where the direction of travel changes, we segment the track's trajectory into segments which are accurately approximated by lines (linear model). Tracklets are determined from the resulting segmentation by creating one tracklet for each segment ("straight" tracklet), as well as one for the path between every two adjacent segments ("turn" tracklet). Furthermore, straight tracklets longer than 100 meters are broken into shorter 50 meter segments. Now, for each tracklet, we compute a collection of attributes, such as location, heading (applies to straight tracklets), heading change (applies to turn tracklets), speed, acceleration, and accumulated distance traveled so far. Geo-registration of input data is a crucial step since tracked objects can be associated with GIS information [7].

3.2 Activity Representation Using ERM

We use ERM (Entity Relationship Models) to capture multiple relationships between elements. Such a framework has been extensively used and validated for a long period of time in real world applications [4, 3]. The basic entity-relationship modeling approach is based on describing data in terms of the three parts: entities, relationships between entities, and attributes of entities or relationships.

We represent track points $\{p\}$, tracklets $\{x\}$, and tracks $\{\mathbf{o}\}$ as entities and link the three entities: $\{p\} \subset \{x\} \subset \{\mathbf{o}\}$. The collection of physical properties of each tracklet is represented as the attributes of the tracklet entity, using a RDBMS table in practice.

We also represent geospatial data (traffic rules, roads, buildings, and areas) the same way. An entity "road" is a collection of road segments and each segment has a set of attributes such as type, name, and speed-limit. Table 1 illustrates our ERM representation.

An activity a_j is defined as a collection of tracklets obeying certain properties:

$$a_j = \{x | x \in \Omega_j, C_j(x) > \theta_j\}, \tag{1}$$

where Ω_j, $C_j(x) \in [0, 1]$, and θ_j represent the relationship associated with the activity, the confidence function and the recognition threshold, respectively. The relationship Ω_j links between the attributes of entities, which include both the physical properties of tracklets and the geospatial data. We can define relationships that are not explicitly represented in the ERM.

For example, *"Speeding"* can be seen as an activity defined by the relationship between the attributes of tracklets (e.g. speed) and geospatial objects (e.g. speed-limit):

$$\begin{aligned} speeding := \{x | \quad & r \in \mathcal{G}_{road}, \\ & x.roadID = r.ID, \\ & x.s > r.s, \\ & C(x.s, r.s) > \theta\} \end{aligned} \tag{2}$$

where $r, r.ID, x.roadID, x.s, r.s, x.pos, r.pos$ represent a road from GIS data ($\mathcal{G}_{road}$), its ID, the road ID of tracklet x, the speed of x, the speed limit of r, the location of tracklet x, and the location of road segment r, respectively. $C(x.s, r.s)$ describes the activity confidence, which increases with the gap between $x.s$ and $r.s$. For instance, the confidence can be defined by the Euclidean distance between the tracklet x and the road segment r: $1/(\|x.pos - r.pos\| + \epsilon) > \theta_1$. The confidence measure is used to ensure the reliability of composite activities as well as offering users a way to tune the system. Note that all activities associated with geo-objects should handle this type of location uncertainty.

3.3 Activity Inference

The ERM-based representation implies that inferring an activity is a search problem to find a subset of tracklets from entire data set, which satisfies certain conditions.

ERM is implemented as a standard RDBMS and we can express set operations by SQL to find an activity from our database. The activity recognition problem is equivalent to sending queries to the RDBMS.

A basic SQL statement has *SELECT*, *FROM*, and *WHERE* clauses: The *SELECT* command specifies the output attributes of entities, *FROM* defines the domain entities associated with the activity, and *WHERE* describes the set of relationships to define the activity and also its confidence. Activity definitions can easily be expressed by SQL statements.

3.3.1 Example I: Simple Activity

Activities associated with motion patterns, such as *"U-turn"*, *"Loop"*, and *"3-point-turn"*, are easily defined and inferred by the ERM framework and its corresponding SQL statements.

Definition. A *"Loop"* is defined as a segmented track where there exist two tracklets $\{x_i, x_j\}$ whose Euclidean distance[1] $\|x_i.pos - x_j.pos\|$ is smaller than the traveling dis-

[1]Each position of a tracklet represents the geometric mean

tance:

$$Loop = \{x_i, x_j| \quad (1 - \frac{\|x_i.pos - x_j.pos\|}{x_j.acc - x_i.acc}) > \theta, \\ i < j, \\ x_i.ID = x_j.ID\}, \tag{3}$$

where $(x_j.acc - x_i.acc)$ represents the traveling distance between x_i and x_j. The traveling distance is computed as the difference of the accumulated distances between these two tracklets.

The above definition is represented by SQL as shown in Table 2, where RDBMS tables T1 and T2 come from the input tracklet table (e.g *SELECT* * INTO T1 *FROM* tracklet) and dist$(\cdot,\cdot)$ is a user defined function [2] to compute the Euclidean distance.

Table 2: SQL: *"Loop"*

```
SELECT * FROM T1, T2
WHERE
      T1.track_id = T2.track_id AND
      (1 - (dist(T1.pos, T2.pos)/(T2.acc - T1.acc))) > θ
```

Note that this definition provides multiple locations of tracklets for each loop due to the inequality constraint. Also, it returns only a set of tracklets that correspond to the starting and ending locations of a loop. Which means that we might need some additional post-processing steps to refine the results. One can extract all tracklets in between the starting and ending locations if the shape of the loop is important. In this paper, we focus on identifying key tracklets.

3.3.2 Example II: Composite Activity

Suppose that we have three independent events identified as three entity sets: *"Entry"* (a_{En}), *"Stay"* (a_{St}) and *"Exit"* (a_{Ex}). *"Visit"* is a composite activity that can be described as a combination of these events.

Definition. We define *"Visit"* as the sequence of a_{En}, a_{St}, and a_{Ex}:

$$visit = \{x_j| \quad i = j-1, k = j+1, \\ x_i \in a_{En}, x_j \in a_{St}, x_k \in a_{Ex}, \\ C(x_i)_{En}C(x_j)_{St}C(x_k)_{Ex} > \theta\}, \tag{4}$$

with x_i, x_j, x_k, three tracklets from the same track. The confidence of *"Visit"* is defined as $C_{En}(x_i)C_{St}(x_j)C_{Ex}(x_k)$ and the confidence of each activity should be normalized in the corresponding SQL implementation.

3.3.3 Example III: Multiple Actors Activity

Activities associated with multiple actors, such as *"Source"*, *"Sink"*, *"Convoy"*, and *"Following"*, can also be defined and inferred by ERM and SQL statements. We identify a source of tracks by finding a set of tracks that have the same starting location in different time periods.

Definition. Let us first define *"2-Source"* as a temporary set of pairs of tracklets which exit from the same location:

$$2src = \{(x_i, x_j)| \quad x_i.trackID \neq x_j.trackID, \\ x_i \in a_{Ex}, x_j \in a_{Ex}, \\ \|x_i.pos - x_j.pos\| < \omega\}, \tag{5}$$

where ω is a threshold. It provides a set of tracklet pairs which appear as many times as they are involved in a 2-tuple source. To extract N-tuple sources, we need to count the number of occurrences of each tracklet:

$$source = \{x_i|S_i = \{(x_i, \cdot) \in 2src\}, |S_i| > \theta\}, \tag{6}$$

where $|S_i|$ is the cardinality of each subset S_i which contains the same tracklets in the pairs of the 2src set. Note that this definition provides all "SOURCE" , where the number of tracklets is greater than two and we can count the number of tracklets as a confidence measure.

3.3.4 Example IV: Geospatial Activities

The ERM framework is ideally suited to incorporate GIS information, as was shown for *"Speeding"*(Section. 3.2). Many activities can only be inferred within the context of geospatial information. We can find "all tracklets on a specific road" by looking at the correspondences between the locations of tracklets and the locations of known road segments.

Definition. *"On-road-X"* is a set of tracklets which are on the same road:

$$on\_road\_X = \{x| \quad x.roadID = r.ID, \\ r.name = \text{“}X\text{”}, \\ 1/(\|x.pos - r.pos\| + \epsilon) > \theta, \\ \forall r \in \mathcal{G}_{road}\}, \tag{7}$$

where r and $r.name$ designate a road segment and its name, and $\|x.pos - r.pos\|$ the Euclidian distance between the road segment and the tracklet. In SQL implementation, we compute the location of each tracklet in advance, and store the id of road segment into the tracklet table. The optional condition $(1/(\|x.pos - r.pos\| + \epsilon) > \theta)$ provides a confidence measure.

Note that most spatial activities can also be enriched by having a geospatial attribute. For instance, a *"convoy"* becomes a "convoy traveling on highway X" when the spatial tracks are associated with geospatial information.

3.4 Scalability

One of the benefits of using an ERM model is that there exist highly optimized RDMBS commercial implementations such as [1]. Furthermore, there has been serious effort in making RDBMS perform equally well in distributed environments, under high load, and with limited downtime. Therefore, by expressing activity definitions in SQL, we can take advantage of existing, distributed, industrial parsers, making our proposed system very scalable.

4. EXPERIMENTAL RESULTS

We have implemented our framework using a standard RDBMS [1], and validated the approach on real visual tracks and GPS datasets. We define 7 activities for evaluation (including *"Loop"*): a three point turn (*"3PT"*) consists of two neighbor turns $\{x_i, x_j|((x_i.\phi/\pi)(x_j.\phi/\pi))/(\|x_i.pos - x_j.pos\|) > \theta\}$; a two point turn (*"2PT"*) has an acute angle: $\{x|x.\phi > \theta\}$; *"Stay"* is defined by the ratio between the time and travel distance between two points $\{x_i|(\|(x_j.time - x_i.time)\|) / (\|x_j.acc - x_j.acc\|) > \theta\}$; *"U-turn"* has an acute angle turn between two tracklets which are located on the same road $\{x_j|x_j.\phi < \pi/4, \|x_i.pos - x_k.pos\| < \omega_1, (x_k.acc - x_i.acc) > \omega_2\}$; *"Entry"* and *"Exit"* are defined with a stop, turns, speed changes and the travel distance. *"Entry"* is defined as $\{x_k|x_k.s < x_i.s, x_k.end = True, x_j.\phi > \omega_1, x_k.acc > \omega_2\}$, where

between the starting and the ending points.

[2]For simplicity, we use an abstract notation and the function can be implemented using a common RDBMS [1].

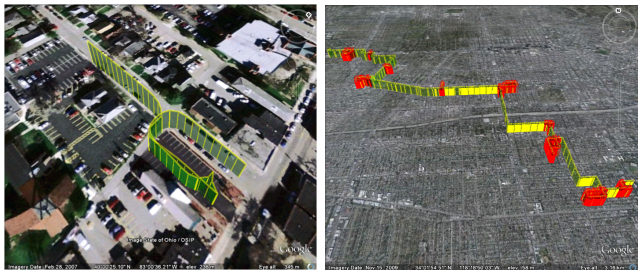

Figure 2: Examples of identified Loops.

$\{x_i, x_j, x_k\}$, $x.\phi$, and ω_i represent different tracklets from the same track, such as $i < j < k$, the turn angle attribute, and internal thresholds associated with the definition, respectively.

4.1 Real dataset (CLIF 2006)

Data. The dataset is a set of tracking results extracted from the CLIF 2006 dataset [2]. This dataset contains wide area motion imagery captured from an airborne sensor. The sensor is composed of a matrix of 6 cameras, where the size of each image tile is 4008×2672. The video is captured at roughly 2Hz, and it is in grayscale. The footprint of the area where we computed tracks is about $1km^2$, and its duration is about 8 minutes. Each track is on average 1 minute long. The total number of tracks estimated in the sequence of interest is more than 8000.

Method. Our input is a set of tracks extracted by the tracking module. To build a set of ground truth data, from a set of automatically extracted tracks, we manually identified individual tracks that include pre-defined activities and assign labels for each data. In our dataset, we identified 2 loops, 2 2-point turns, 3 3-point turns, 8 entry and 7 exits. We inserted all identified tracks into a single table in our RDBMS and inferred activities using pre-defined SQL statements.

Some tracks have more than one activity (e.g., a loop and a 3 point turn) but the locations associated with specific activities can be different. To evaluate the result of an activity, we extracted all tracklets, compared to the activity definition, from entire dataset, visualize the result, and then, verify manually whether the extracted tracklets represent the actual activity or not.

Results. We were able to identify all simple activities, such as "2 point turn", "3 point turn", "Entry", "Exit", and "loop", which can be easily seen in real data set. The precision and recall were 0.76 and 0.86, respectively.

In addition, we identified a number of geospatial activities, such as "on road X", "speeding", and "approaching X", as well as some complex activities including multiple actors, such as "source (or sink) around X". The extracted activities and geospatial objects can be visualized using Google Earth, where we can identify both activities and associated geospatial objects. Figure 2 (left) shows one of identified geospatial activities in the real dataset.

4.2 GPS trajectory dataset

Data. We also evaluated our method on labeled data from GPS. The data was acquired from cooperative users' GPS units in Los Angeles. We used a standard GPS to record short trips between 10 and 40 minutes long. GPS filters were deactivated, so only raw data have been recorded. Compared to the visual tracks obtained from our tracking module, GPS tracks do not differ a lot. First the localization error is similar to video geo-registration. Second, the GPS acquisition frequency (1Hz) is only half our video framerate (2Hz).

Method. We use the same tracklet segmentation module to extract tracklets from the GPS dataset. To build a set of ground truth data from a set of automatically extracted tracks, we manually selected individual tracks that include pre-defined activities and assign labels for each data. The dataset includes 17 loops, 7 three point turns, and 13 u-turns.

Results. We were able to identify all simple activities, such as "Loop", "3-point turn", "U-turn", and "Stay", which can be easily seen in real data set. The precision and recall were 0.97 and 0.90, respectively. Fig. 2 (right) shows a set of identified activities ("Loop") in a single GPS track.

5. CONCLUSION

Our results show that using Entity Relationship Models enables us to identify simple activities, such as "U-turn", as well as some complex activities including multiple actors, such as "source" and "sink" in wide area aerial imagery.

In addition to further validation, we will work on finding simpler mechanisms to define events and verifying efficiency. Also, we plan to incorporate probabilistic reasoning into our model to better handle uncertainties in the data.

Acknowledgment

This work was supported in part by grant DE-FG52-08NA28775 from the U.S. Department of Energy.

6. REFERENCES

[1] http://www.microsoft.com/sqlserver/.
[2] http://www.sdms.afrl.af.mil/index.php?collection=clif2006.
[3] P. P. Chen. The entity-relationship model - toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36, 1976.
[4] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.
[5] H. Gupta, L. Yu, A. Hakeem, T. E. Choe, N. Haering, and M. Locasto. Multimodal complex event detection framework for wide area surveillance. In *CVPRW*, pages 47–54, 2011.
[6] R. Lange, F. Dürr, and K. Rothermel. Scalable processing of trajectory-based queries in space-partitioned moving objects databases. In *16th ACM SIGSPATIAL*, GIS '08, pages 31:1–31:10, New York, NY, USA, 2008. ACM.
[7] Y. Lin, Q. Yu, and G. Medioni. Efficient detection and tracking of moving objects in geo-coordinates. *Mach. Vision Appl.*, 22(3):505–520, May 2011.
[8] E. Pollard, B. Pannetier, and M. Rombaut. Convoy detection processing by using the hybrid algorithm (gmcphd/vs-immc-mht) and dynamic bayesian networks. In *FUSION '09.*, pages 907–914, 2009.
[9] A. Pozdnoukhov. Dynamic network data exploration through semi-supervised functional embedding. In *Proceedings of the 17th ACM SIGSPATIAL*, pages 372–379, 2009.
[10] J. Prokaj, M. Duchaineau, and G. Medioni. Inferring tracklets for multi-object tracking. In *CVPRW*, pages 37 –44, 2011.
[11] V. Reilly, H. Idrees, and M. Shah. Detection and tracking of large number of targets in wide area surveillance. In *ECCV (3)*, pages 186–199, 2010.