A Framework for Activity Detection in Wide-Area Motion Imagery

Reid Porter*, Christy Ruggiero, John D. Morrison

Los Alamos National Laboratory, Los Alamos, NM, USA.

ABSTRACT

As wide-area persistent imaging systems become cost effective, increasingly large areas of the earth can be imaged at relatively high frame rates. Efficient exploitation of the large geo-spatial-temporal datasets produced by these systems poses significant technical challenges for image and video analysis and for data mining. Significant progress in image stabilization, moving object detection and tracking, are allowing automated systems to generate hundreds to thousands of vehicle tracks from raw data, with little human intervention. However, tracking performance at this scale is unreliable, and average track length is much smaller than the average vehicle route. These are limiting factors for applications that depend heavily on track identity, i.e. tracking vehicles from their points of origin to their final destination. In this paper, we propose and evaluate a framework for wide-area motion imagery (WAMI) exploitation that minimizes the dependence on track identity. In its current form, this framework takes noisy, incomplete moving object detection tracks as input, and produces a small set of activities (e.g. multi-vehicle meetings) as output. The framework can be used to focus and direct human users and additional computation, and suggests a path towards high-level content extraction by learning from the human-in-the-loop.

Keywords: Activity detection, tracking, video, wide-area motion imagery.

1. INTRODUCTION

Recently, wide-area airborne imaging sensors have come into practical use. These systems image small city-sized areas at approximately 0.5 meters per pixel and about 1 or 2 frames per second. Due to the wide field-of-view and long dwell times (hours to weeks), these collection systems allow for the observation of many dynamic phenomena that were previously inaccessible in satellite or street-level video imaging systems. In particular, wide-area motion imagery (WAMI) provides the opportunity to track a large fraction of the vehicles within an urban environment scene from their point of origin to their final destination. These tracks can then be used to extract higher-level information, such as vehicle activities. Examples of vehicle activities include the general classification of vehicle routes into categories such as commuter, commercial, or tourist, but can also include many specialized classifications relevant to particular applications such as delivery, get-away, or surveillance. Activity detection has many applications including tactical scenarios in defense applications, real-time emergency response, and urban planning.[1]

Activity detection in WAMI has some similarity with activity detection problems faced in surveillance and security systems that use multiple, fixed, high frame-rate, narrow field-of-view video cameras[2,3,4]. Specifically, both datasets have persistent data collection, which allows systems to build and exploit statistical models of normal behavior over time, and both systems have a fixed frame of reference, which means models of the observable area can be used to provide contextual information relevant to many activities of interest[5]. WAMI activity detection also has several unique characteristics, which make it a new, interesting, and challenging research problem. In particular, airborne collection platforms bring a distinct set of challenges, such as geo-registration and parallax; lower levels of spatial and temporal resolution; and, the predominant objects and scales of interest (vehicles driving across a city), have different dynamics and associated activity characteristics. In addition, vehicle activities typically exist within the structured environment of a road network[6,7] and within a much larger and richer contextual background of a geo-spatial information system, which can be both an advantage and disadvantage.

* {rporter, ruggiero, jmorrison}@lanl.gov

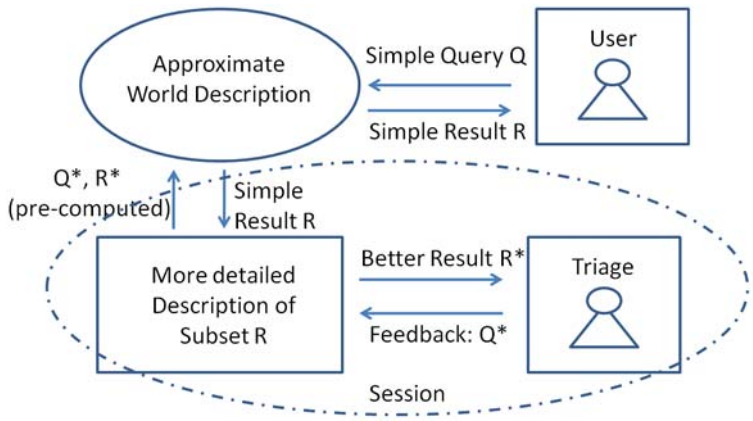

Figure 1. Content based search via interactive refinement of a world description.

These differences cause WAMI vehicle activity detection to be more critically dependant on obtaining complete and accurate vehicle tracks compared to narrow-field, high frame-rate video camera applications. In addition, its unique spatial and temporal resolutions mean accurate tracking is much more difficult for WAMI[7]: 1) *point-like* moving objects (vehicles) move anywhere from 1 to 200 pixels, 2) parallax introduces large amounts of motion clutter due to oblique viewing angles, and 3) registration is often required in real-time and is therefore approximate, e.g. stationary objects might appear to move up to 10s of pixels. With such challenges, any practical activity detection system must be intimately related to the tracking system. The primary contribution of this paper is a framework for vehicle activity detection that can work with existing WAMI tracking systems, and over time, refine the representation of vehicle routes to improve subsequent activity detection. Our approach is similar in spirit to the joint event detection and track matching approach presented in Ref. 4, but is unique to the activities and tracking problems associated with WAMI.

In Sections 2, 3 and 4 we describe the general technical approach and system components that we have implemented. Section 5 describes a basic set of activity detection tools that we propose as an *initial* basis for a large class of activity detection applications and Section 6 describes our experimental evaluation of one of these activity detection tools. We conclude in Section 7 with a discussion of how the framework can be extended and improved, and promising directions for future research.

2. CONTENT BASED SEARCH OF WIDE AREA MOTION IMAGERY

The traditional approach to intelligent searching of complex image and video datasets is through a feed-forward video exploitation pipeline. This pipeline translates raw pixels into a high-level representation (a *world description*) that can be easily searched with traditional SQL (Structured Query Language) type queries. The feed-forward pipeline for WAMI exploitation is similar to many other video exploitation systems, and some of the components are described in Section 3. Typically this pipeline is tuned to a particular design point in the performance/cost trade space (as good as we can afford) and produces a fixed stream of meta-data that is stored in a database for future search. This approach to intelligent search presents a number of problems for activity detection in WAMI. First, because activities are spread over large space and time, there is substantial variability in the data quality even within a single activity. Second, because tracking quality is highly variable over space and time, it is almost impossible to decide what level of performance is sufficient. For example, busy multi-lane intersections can require orders of magnitude more computation to resolve to the same level of precision as single lane intersections with less traffic. These differences in required precision motivate the guiding concept behind our activity detection framework: the performance/cost design point of the tracking system is dynamic and is intimately linked to the activity detection queries made by the user.

The main components of the proposed framework are shown in Figure 1 which we describe through example. An approximate world description is generated with a traditional feed-forward tracking pipeline at relatively low cost. This typically produces a large number of track segments (10-100 segments per route), and a large number of spurious tracks. A user makes an activity detection query (Q) – for example, *"find all vehicles that took a specific exit ramp to a specific shopping centre"*. A query response (R) returned by a traditional SQL query would be very poor since it is likely that no vehicles were tracked successfully between these two distant locations. Instead, our framework uses the query to select the relevant subset of the world-description and tracking model. This subset is then refined by using more computation

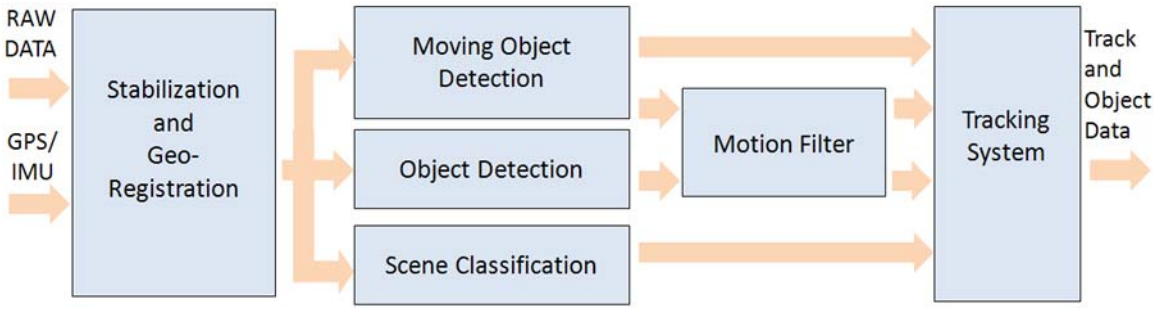

Figure 2. Feed-forward WAMI exploitation pipeline used to produce the world description.

to obtain better tracks, and interaction with the user to resolve ambiguities and validate results. This process is user intensive, but produces improved results (R*) that also can be used to refine the activity query over time (Q*), so that similar queries in the future are less user-intensive. In future work we also suggest that by observing high demand for particular queries, the feed-forward pipeline can selectively include components at different points in the performance/cost design space. That is, if a large number of the end-users all require the same world description refinements, then these results should be accessible through simple queries. In the next few sections we provide specific examples of how Figure 1 is implemented for WAMI.

3. APPROXIMATE WORLD DESCRIPTION

The feed-forward pipeline used to generate the initial vehicle track segments is very similar to other video exploitation pipelines and is shown in Figure 2. The first step involves stitching together images from multiple cameras, frame-to-frame stabilization and geo-registration. This is a nontrivial, computationally intensive, step for airborne video. The second step is moving object detection, in which, through statistical modeling, each pixel is predicted to be either part of a moving object or part of the background[9]. To improve tracking performance, and to track vehicles through stops, we can apply appearance based object detection[8]. Appearance based scene classification can also be used to dynamically update geo-spatial information systems and for identifying road networks and correct parallax affects of buildings. Finally, our system includes a motion filter for reducing appearance based object detection false alarms and for generating velocity estimates for moving objects[10].

The most important component of Figure 2 for this paper is the tracking system. We briefly outline some of the design choices for WAMI tracking systems and describe the choices made for the activity detection experiments. Multi-vehicle tracking involves the interaction of:

- Variables $X_t = [X_{1,t}, X_{2,t}, \dots, X_{N(t),t}]$ which typically includes the predicted position and velocity of $N(t)$ predicted vehicles in the scene at time t.
- Observations $Y_t = [Y_{1,t}, Y_{2,t}, \dots, Y_{M(t),t}]$ which typically includes the observed position and velocity of $M(t)$ vehicles in the scene at time t.
- A transition model $P(X_t|X_{t-1})$, which simulates the vehicle trajectories.
- A likelihood function $P(Y_t|X_t)$ which relates the observations to the variables.

Our tracking system component must provide $P(X_t|X_{t-1})$, $P(Y_t|X_t)$ and choose a solution method. The most common $P(X_t|X_{t-1})$ involves the independent propagation of each vehicle through time with constant velocity. While this is the model used in this paper, more sophisticated models that have been shown to improve performance include: 1) interacting multiple models in which each vehicle has a mode variable indicating constant velocity, turn or stop, 2) multi-vehicle models in which vehicle motion is constrained to prevent collisions with other cars, and 3) models that are dependent over space and time and constrain vehicle motion based on the structural constraints of the road network. The choice of $P(Y_t|X_t)$ and solution method must jointly address the association problem. This problem arises because we typically do not have unique observables for each vehicle which makes it difficult to *associate* particular observations $Y_{j,t}$ with the correct state variables $X_{i,t}$. The optimal solution requires consideration of all possible assignments, which grows exponentially in time.

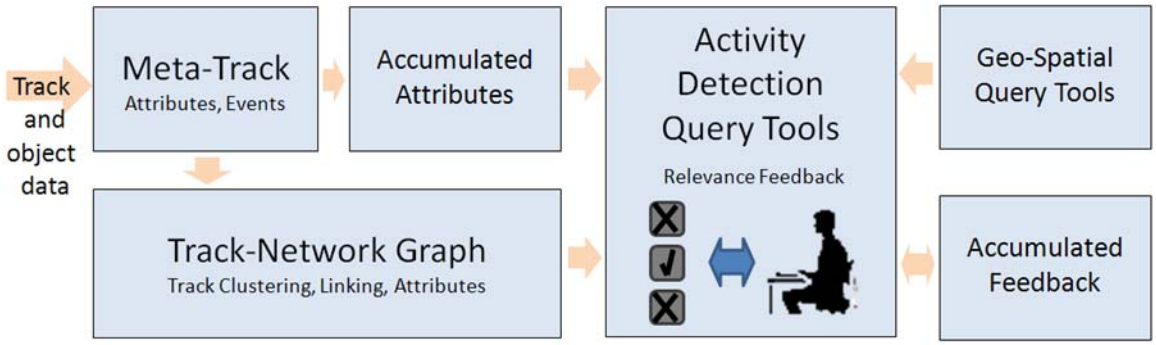

Figure 3. Activity detection system used to interactively refine the world description.

A common solution to the association problem is to simply choose a good association at each time-step. A simple way to choose a good association is to associate $X_{i,t}$ with the closest observation. This method is greedy (decisions are not revisited) but is fast and simple to implement and is the method used to generate the initial world-description in this paper. If multiple variables (vehicles) compete for an observation then we leave the observation unassigned, as suggested in Ref. 4. A slightly less greedy approach to finding a good association is to solve the assignment problem. This approach simultaneously finds the assignment between all variables and observations that minimizes the sum of distances. This approach can be extended to multiple frames leading to a type of multiple-hypothesis tracking (MHT). It can also be used as a sub-routine within the Virtibi algorithm to propagate a collection of high probability associations through time, until evidence accumulates to choose the best associations[11]. As we describe in the next Section, we will use the assignment problem to selectively improve the initial tracking results based on activity detection requests from the user.

4. ACTIVITY DETECTION SYSTEM

The principal components of the activity detection system are illustrated in Figure 3. The primary inputs are track segments generated by a greedy, moving-object tracking system (Figure 2, Section 3). Track segments are sequences of tuples {location X, velocity V and time T} predicted to be generated from the same vehicle. Future systems may include additional information to help improve activity detection significantly. For example using only tracks generated from moving object detection means it is not possible to differentiate between tracks that end because the vehicle left the field of view from tracks that end because the vehicle stopped for brief period of time, such as, at an intersection. Additionally, the current representation incorporates only one prediction for each vehicle for each time period. With Multiple Hypothesis Tracking (MHT), there would be multiple predictions for each vehicle. While we have prototyped the current framework using basic moving object based tracks as input, because the framework can be easily extended to include these additional sources of information as they become available, we discuss these extensions throughout the paper.

4.1 Meta-Track and Accumulated Attributes

The computational cost of activity detection can be reduced at minimal cost in performance by reducing the number of tuples used to represent track segments. That is, the typical WAMI tracking system produces samples every 0.5s, but many of these samples are redundant and can be discarded, e.g. the vehicle is moving with constant velocity or is stopped at an intersection. Using the Douglas-Peucker line approximation algorithm, we found that at an error of 3 pixels (approximately equivalent to stabilization error of our dataset), the number of tuples in the track segment could be reduced by an order of magnitude. The number of tuples in a track segment after this step is typically less than 10.

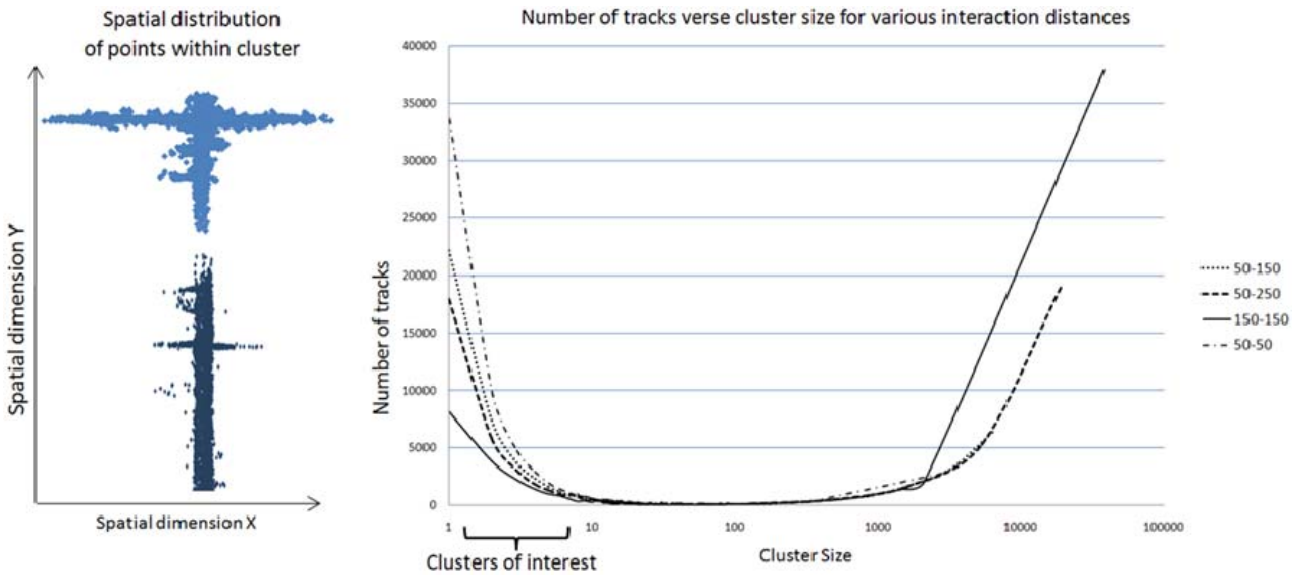

Figure 4: Left) Location of track end points within the largest and second largest clusters accumulated over time and Right) Number of tracks in clusters of various sizes for various values of interaction-distance (pixels - seconds).

The second component of the activity detection system uses track segments accumulated over the entire dataset (or multiple datasets over the same geographic location) to build spatial, temporal and track-centric models of normalcy. For example, the total number of tracks, or track velocity, accumulated as a function of location. These attributes are available through the query system for users interested in activity that is unusual for a given location or time. Our framework applies a number of standard statistical techniques to develop these normalcy models, including a simple count, as well as Gaussian and histogram parameterizations.

4.3 Track Network Graph

The track network graph is one of the most important and useful components of the activity detection framework and provides the main data-structure upon which activity query tools operate. The first step in forming the track network graph is to cluster track segment start and end points. For moving object tracks, the start and end points are simply the first and last location from each track segment. We cluster start and end points independently and use a deterministic clustering method based on an interaction-distance parameter. This distance parameter, provided by the user as part of the query, represents the space (meters) and time (minutes) within which the user believes vehicles may be interacting, e.g. vehicles that stop within 10 meters of each other within 5 minutes. Using a minimum distance clustering method allows clusters to have variable size and clusters can grow arbitrarily large. On the left in Figure 4 we show examples of the two large clusters found in our dataset. The spatial distribution captures some of the structure of the road network; this is due to the fact that busy intersections have the highest density of vehicle stops, leading to large clusters.

On the right in Figure 4, we show the number of track end-points as a function of cluster size for various values of the interaction-distance parameter. This plot illustrates how we can use track clustering to filter large portions of the world description; specifically, because many activities of interest involve a small number of vehicles, we can simply ignore large clusters. As shown in Figure 4, there is a sensitivity to the choice of interaction-distance, and so care must be taken when discarding clusters. In addition, a range of cluster sizes beyond the number of vehicles specified in a query should be analyzed to account for spurious and missed tracks within clusters. An interesting property of track clusters is that large clusters are typically associated with difficult regions for activity detection. For example, large clusters form at fast food drive-throughs, and multi-lane intersection. Since multiple vehicle stops in these regions is a normal activity, detecting specific meetings is practically impossible at the spatial and temporal resolutions of current WAMI systems. Therefore, in the absence of additional sources of information, this also suggests large clusters should be ignored and activity detection resources directed elsewhere.

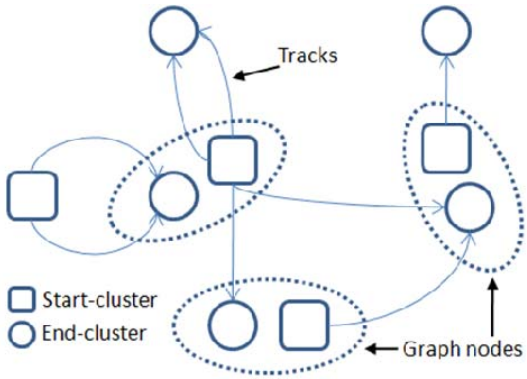

Figure 5: Graphical representation of the track network graph

Using independent start and end point clusters, we form the track network graph, which is illustrated in Figure 5. Track segments provide high confident links between start and end clusters. However links between end and start clusters must be predicted. Consider the case in which a vehicle comes to a stop at an intersection, and then departs several seconds later. For moving object based tracking, this corresponds to two track segments. These segments are associated with end and start clusters, and we would predict a link between these clusters if the end cluster is relatively close in space to the start cluster, and if the start cluster occurs latter in time than the end cluster.

Within the track network graph, nodes represent potential (probabilistic) relationships between clusters as indicated in Figure 5. Because nodes are probabilistic, clusters can be associated with multiple nodes. The concept of probabilistic nodes is key to robust activity detection: a node localizes the uncertainty in track identity that was not resolved during tracking, and hence, allows activity detection to be performed on noisy and incomplete tracks. Note, we do not have to consider all possible nodes that link every end cluster with every start cluster. As described in the example, spatial and temporal constraints associated with the activity of interest are used to reduce the set of candidate nodes. The same interaction-distance parameter is used to cluster vehicle starts and stops to determine if start and end clusters are close enough in space and time to be considered a node.

The motivation for the track network graph is most obvious for overcoming limitations in moving object tracking since track ends and track starts often correspond to a vehicle that stops and then starts. However, this concept also generalizes to more complex tracking models (described in Section 3) with minor modifications. For example, for appearance based tracking systems that track through stops, the track start and track end-points are not necessarily associated with vehicle stops; rather, they are often associated with low track confidence. For activity detection algorithms the performance enhancements provided by this approach will be different (and easier), but the key point is that nodes still localize track uncertainty.

5. ACTIVITY DETECTION QUERY TOOLS

Activity detection tools are user-configurable queries on the track network graph. Several graph attributes relevant to the activity of interest are calculated and used to rank nodes, paths, and sub-graphs that correspond to the activities of interest. Note that for efficiency, attributes are typically calculated only when required by specific queries. This is because other query parameters, such as cluster size (number of vehicles), normalcy models, and geo-spatial attributes can used to filter large parts of the track network graph that are not relevant to the activity of interest. We suggest a loose basis for WAMI activity detection that covers a large number of activities of interest, and includes activities that are often sub-components of more complex activities. These three categories are illustrated in Figure 6 and include queries on:

1. **Nodes:** these activities correspond to multiple vehicle stopping and starting behavior (e.g. meetings) within a relatively small amount of space-time.
2. **Paths:** these activities are typically associated with a single vehicle and are related to the vehicle behavior with respect to normalcy models, geo-spatial attributes (e.g. anomalous routes).
3. **Sub-graphs:** these activities are associated with multiple vehicles driving similar or related routes (e.g. coordinated driving).

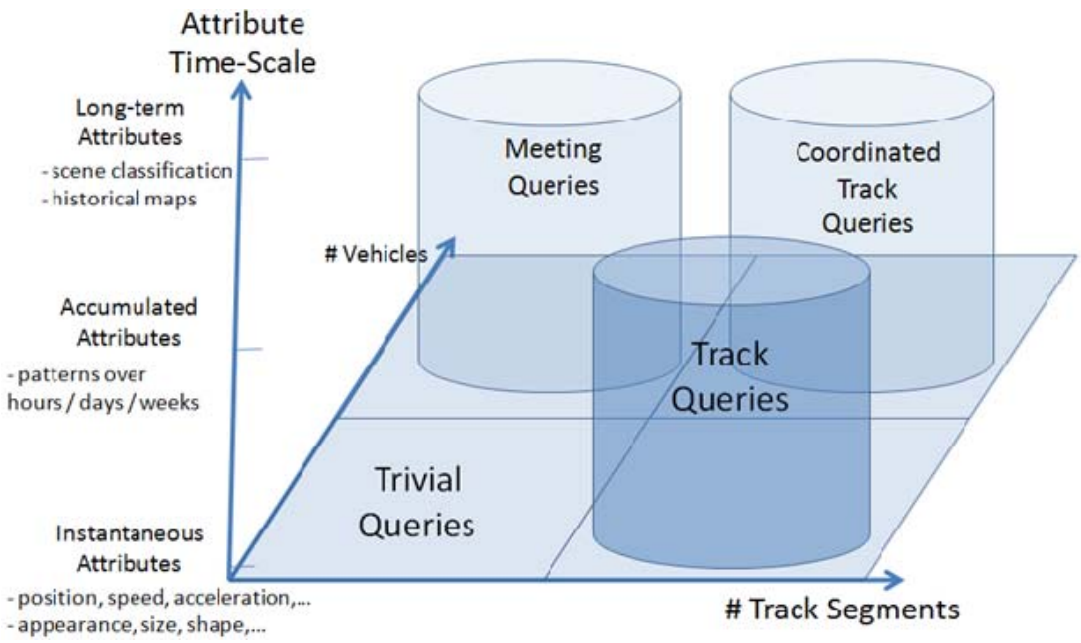

Figure 6: An initial activity query basis for WAMI

5.1 Meeting Detection Queries

The main technical challenge that we address is how to provide useful activity detection with current tracking performance. Our initial prototype, and the focus of this paper, is the meeting detection tool. We define meetings as locations where two or more vehicles stop and/or start in close proximity. These locations sometimes correspond to real meetings, but more often, it is a location where tracking was unreliable or failed. By providing a tool that robustly identifies actual meetings verse meetings due to track error, we not only identify interesting activities, we provide a mechanism to correct and account for track error. Therefore, this tool provides a basis for developing robust track and coordinated driving query tools.

The key node attribute for the meeting detection tool is the ***Association Cost***. Given two sets of points from the end and start clusters, we calculate the association cost of a node by solving the assignment problem described in Section 3. This cost is zero if the end and start points are equal in number, and in identical positions. This technique is similar to how we improve tracking prediction, but in our case it is only applied to nodes that are relevant to the user query. Another node attribute is the ***Meeting Duration***, which measures the time between the last point in the end cluster and the first point in the start cluster, e.g. length of a potential meeting. A different group of attributes are calculated independently for end and start clusters. These include: ***Compactness***, the variance in spatial location of points within the cluster; ***Direction***, the variance in velocity associated with points within the cluster, and ***Arrival / Departure Duration***, the difference between the maximum and minimum time for points within the cluster.

These attributes are selected by the user within a query using an associated weight. For user-convenience, we developed a number of preset configurations for the attributes and weights that were found to work well in practice. In future work these attributes and weights can be optimized using machine learning techniques. The two key attributes that were found to most help identify legitimate meetings were the association cost and a normalcy model for stops in the area – probability of a stop at that location. Combined, these attributes provide a query that effectively detects multi-vehicle meetings and is extremely robust to noisy, missing and false tracks. Nodes that are predicted as meetings typically include one start and one stop cluster; however, it is also useful to consider nodes with a single start, or stop cluster (e.g. end clusters) – those that do not have a start cluster which is sufficiently close. These nodes might correspond to a meeting where the arrival was observed but the departure was outside of the collection window. This possibility can also be selected by the user; however as expected, the quality of meeting detection is typically worse.

5.2 Track and Coordinated Driving Queries

The final goal of the proposed framework is to detect activities of interest that occur over larger space/time dimensions than any one of the input track segments. We propose that the meeting detection query will play an important role in realizing these more complex queries. The key concept is that meeting detection provides a refined association probability that can be used to accumulate query attributes through paths and sub-graphs. The association cost from the assignment problem is the key attribute in our prototype; however, we can also exploit any other attribute information that is found useful during meeting detection. In addition, meeting detection may also include multiple associations through which path and sub-graph attributes can be accumulated, much like multiple hypothesis tracking. How the proposed methods will scale with the number of track segments is an open question; however, we propose our approach must be at least as efficient as the traditional approach of enhancing the feed forward tracking performance, and may be much better since tracking improvement is selectively applied.

Additional attributes that could be accumulated over paths and sub-graphs for track and coordinated driving queries include measuring quantities between track points and also between two different tracks that are relatively close in space and time. We use the reduced meta-track representation (less than 10 points / track) to produce a N by M characterization matrix, where N and M are the number of points. The matrix is populated by applying a function to every pairwise combination of points i.e., the $(i, j)^{th}$ element is a function of ith and j^{th} track points. This N^2 representation is required to characterize track interactions at all scales. Some examples of pairwise functions that are may be of interest include ***Distance / Track Length*** for single tracks to characterize inefficient routes, and ***Relative Distance*** for multiple tracks to characterize vehicles driving in formation. Once the characterization matrix has been calculated, an additional function is applied to produce a small set of scalars used in the queries. The function depends on which pair-wise function is used but is typically a maximum or minimum.

5.3 Geo-spatial Query Tools and Accumulated Feedback

The geo-spatial and accumulated feedback components of the framework extend the activity detection query system to include geo-spatial context information, and user derived domain knowledge. Geo-spatial attributes are an essential component of any practical WAMI activity detection, so efficient interfaces between spatial and the WAMI moving object data types will be required. Scene characterization, as described in the world description, can also play a role here, and can provide more timely geo-spatial information. This is an ongoing topic of research in geo-spatial information systems.

Accumulated feedback is an area where we see great opportunity with the proposed framework. At a simple level, accumulated feedback includes query triage, in which the user provides labels to the ranked list of activities returned by the query tools. A more ambitious role for accumulated feedback appears when one considers longer term data exploitation needs. Since WAMI is associated with an absolute geographic coordinate system, and since the amount of data and number of users and applications over a given area is growing rapidly, there is also a growing opportunity to exploit tool usage statistics. Similar to what has been observed in the internet search domain, we suggest that accumulating the queries, results, and triage statistics from many users over long periods of time will become as important as the activity detection tools themselves. These statistics can not only refine and improve query results, but also be used to tune the framework towards particular activities that are most relevant to specific applications and user communities.

6. MEETING DETECTION TOOL EVALUATION

An important motivation for the proposed framework is to provide practical tools of immediate value to real world users. This not only validates the approach, but also lays the foundation for accumulating user feedback, which we propose is key to the long term success of WAMI exploitation. To test the practical utility of our approach, we performed an extensive test of the meeting detection tool on several real-world WAMI datasets. The user's task for the experiment was to detect two-, three- and four-car meetings. In the test, the user used the meeting detection tool to select prioritized "batches" of likely candidates. Because larger batches included an increasing number of false targets, which the user was required to remove, the test determined the difference between the time required to delete the false targets from a single or exhaustive set of prioritized batches provided by the tool, against the time required to find the same meetings using a manual approach i.e. unaided visual search of the entire dataset.

Table 1. Summary of meeting detection results obtained from 4 WAMI datasets containing ground-truthed meetings.

Collection Area	Coverage Time (min)	Ground-Truthed Meetings	Candidates Considered By the User (Initial batch size)	Total Meetings Identified	User Time with Tool (Time to eliminate False Alarms)	Detection Rate (% of Meetings found in initial batch)	Total Analysis Time required for 100% detection (Estimate)	Manual Analysis (24hr day)
1	136	16	500	84	90 min	0.5	10 hours	7.5 days
2	48	5	250	9	45	0.6	2 hours	2.5 days
3	43	4	250	19	45	0.0	7.5 hours	2.5 days
4	46	5	250	11	45	0.2	NA	2.5 days

The test results are summarized in Table 1. The first four columns reflect the characteristics of the WAMI data set used for the test. Columns 5-8 provide the experimental results. Column 9 is an informed estimate of the time required to conduct an equivalent search using visual inspection (20 frames per second, or 10x real-time, with no stops) – this estimate was subsequently verified as optimistic by WAMI end-users. The difference between the times in Column 8 and 9 represents the efficiency that the Meeting Detection Tool provides analysts searching for meetings. As an example, in Row #1 in Table 1, the user selected an initial batch of the 500 most promising meeting candidates from the thousands of possible meetings that the activity tool found. It took the analyst 90 minutes to review the 500 candidates and eliminate 400 false positives. The 84 meetings found included half of the ground-truth meetings in the data along with 68 meetings that were not involved in the data collection. It took an additional 8.5 hours to find all of the 16 ground-truth meetings.

Figure 7 illustrates, for four different data sets within the experiment, the rate of detection as estimated from the user triage of the initial batch size. Curves on this plot indicate the rank of the remaining ground-truth meetings within the prioritized list generated by the query tool. The differences in the rate between the four data sets reflect differences in the terrain and traffic characteristics of the four collection areas. The data reflects an improvement in search time of a factor between 3 and 10. For two of the 4 datasets, the meeting detection tool appears to provide substantial increase in productivity. Using our tool, the user is able to find about 75% of the meetings in about 2 hours. This should be compared to an optimistic estimate of 3 to 6 days for the unassisted user.

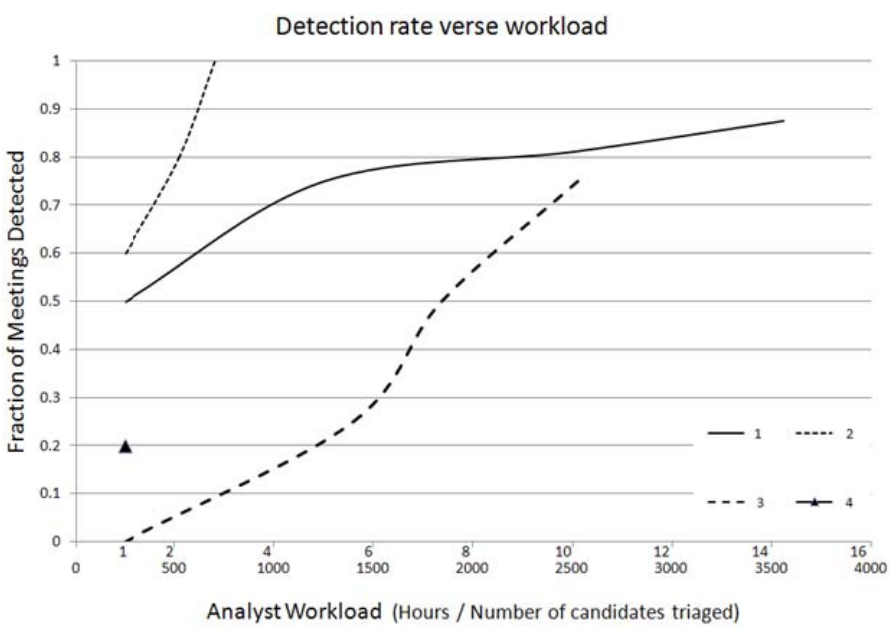

Figure 7: Rate of meeting detection for the 4 WAMI datasets.

7. SUMMARY

To detect and characterize activities that operate over large space and time, it is essential to maintain track identity over longer proportions of a vehicle route. It is not likely that practical tracking systems will achieve this objective in the foreseeable future. Therefore, to detect useful patterns of activity we suggest a framework in which we will build increasingly complex classifiers using a graphical representation of noisy and incomplete track data. We have developed, prototyped and tested the first activity detection tool for this approach – a node classifier that can be used to detect multiple vehicle meetings. This classifier improved user productivity by a factor of 18 on a realistic WAMI exploitation problem. It is also provides the key component of a general framework for activity detection that can simultaneously detect complex activities while predicting high probability routes through multiple track segments.

ACKNOWLEDGMENTS

A large number of people contributed to this work in a large number of ways. We would like to acknowledge and thank Steven Brumby, Mathew Fair, Andy Fraser, Neal Harvey, Don Hush, Paul Pope, Lakshman Prasad, Alexei Skurikhin and James Theiler from Los Alamos and Anthony Hoogs from Kitware Inc. for many useful technical discussions. We would also like to thank Wayne Scoggins and David Aubrey from Los Alamos as well as Kitware Inc. for valuable software development.

REFERENCES

1. B. Gillis, R. Guensler, C. Grant, “Collection of Vehicle Activity Data by Video Detection for Use in Transportation Planning”, *Journal of Intelligent Transportation Systems*, **5**(4), pp. 343-361, 2000.
2. C. Stauffer and E. Grimson, “Learning patterns of activity using real-time tracking”, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, **22**(8), pp. 747-757, 2000.
3. Niu, J. Long, D. Han, and Yuan-Fang Wang, "Human Activity Detection and Recognition for Video Surveillance", in *Proceedings of the IEEE Multimedia and Expo Conference*, Taipei, Taiwan, pp. 719-722, 2004.
4. M. T. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, G. Doretto, “Joint Recognition of Complex Events and Track Matching”, *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 1615-1622, 2006.
5. C. L. Lin, Q. Zheng, R. Chellappa, L.S. Davis, X. Zhang, “Site model supported monitoring of aerial images”, in *Computer Vision and Pattern Recognition*, pp. 694-699, 1994.
6. E. Swears, A. Hoogs, and A. G. A. Perera, "Learning Motion Patterns in Surveillance Video using HMM Clustering", in *Proceedings of the IEEE Workshop on Motion and Video Computing*, Copper Mountain, CO. 9 Jan 2008.
7. R. Loveland, E. Rosten, R. Porter, “Improving multiple target tracking in structured environments using velocity priors”, *Proc. SPIE*, Vol. 6969, 2008.
8. R. Porter, R. Loveland, E. Rosten, “Building Robust Appearance Models using On-line Feature Selection”, *Proc. of the SPIE*, Vol. 6574, pp. 657409, 2007.
9. R. Porter, N. Harvey, J. Theiler, “A Change Detection Approach to Moving Object Detection in Low Fame Rate Video”, *Proc. of the SPIE*, this volume, 2009.
10. R. Porter, A. Fraser, R. Loveland, E. Rosten, “A Recurrent Velocity Filter for Detecting Large Numbers of Moving Objects”, *Proc. of the SPIE*, Vol. 6969, pp. 69690C-69690C-9, 2008.
11. A. Fraser, “Model Based Tracking”, *Los Alamos National Laboratory Technical Report*, C-08025, 2008.