Huerta, Schade, Granell (Eds): Connecting a Digital Europe through Location and Place. Proceedings of the AGILE'2014 International Conference on Geographic Information Science, Castellón, June, 3-6, 2014. ISBN: 978-90-816960-4-3

Real-time detection of anomalous paths through networks

Steven D. Prager
University of Wyoming
Department of Geography
Laramie, USA
sdprager@uwyo.edu

R. Paul Wiegand
University of Central Florida
Institute for Simulation & Training
Orlando, USA
wiegand@ist.ucf.edu

Abstract

The proliferation of increasingly inexpensive mobile devices capable of transmitting accurate positional information to other devices and servers has led to a variety of applications ranging from health situation monitoring to GPS-based offender monitoring. One of the resultant challenges is in understanding, in real-time, when incoming observations merit further examination. In this research, we investigate an approach for identifying anomalous paths through networks using real-time comparisons to a previously learned model. Our approach, the development of a series of "posterior weighted graphs" allows us to both determine which underlying model a particular path most closely represents as well as evaluate this relationship in real-time as more observations become available. Here we present the posterior weighted graph approach for examining path similarity and an extension for detecting anomalies in real-time. Our results illustrate how we can distinguish from among multiple candidate paths and, likewise, when observations no longer match an expected model.

Keywords: path similarity, anomaly detection, networks, mobility, GPS

1 Introduction

One challenge in understanding paths through networks is detecting when observed paths depart from what is considered normal. What is normal is, of course, subject to *a priori* establishment of corresponding expectations. In this paper we present an approach for learning an *a priori* model for a set of potential paths. We then demonstrate how this model can be used to facilitate real-time detection of when observed paths depart from the expected path(s) represented by the learned model. Applications of real-time path anomaly detection range from the health field [7] to fraud detection [3], to automated surveillance of individuals, traffic, objects and crowds [9].

In the context of this study, we define anomalous event as an event that has characteristics significantly different than normal [9]. Proliferation of track data from mobile devices has led to a variety of applications wherein the goal is to detect anomalous mobility patterns [2, 7]. In such cases, the anomaly occurs when an observed mobility pattern departs from a previously established pattern. Often couched in terms of "path matching" problems [6], many methods are used to look at path similarity [4].

The challenge of working with similarity detection methods for real-time path data is that the paths and, hence, the corresponding metrics are constantly changing [5]. Similarly, there are a potential for a number of ambiguous cases [8]. Alternatively, it is possible to classify a dynamic path against an established baseline [1, 3]. In both [3] and [1], a baseline is established with previously collected GPS traces. Though [3] uses a grid-based approach in conjunction with isolation-based methods and [1] uses a reduced "support point" representation, both compare emerging trajectories to a previously established baseline.

Here, we present a method capable of using either previously collected GPS data or baseline paths from a map interface such as Google Maps. In turn, we present a new method for discerning departures from this baseline using a series of weighted graph models. In the next section we address the problem and, following, illustrate the methods and analytic results.

2 Problem Definition

The principle emphasis of this research is to determine whether an observed path departs from an expected path and to make this determination in real-time.

Consider a street network represented by a series of nodes and edges. Paths through that network can be represented as a collection of ordered vertices where, by extension, traversal of a vertex implies traversal of the corresponding edge between a vertex and the previous vertex. Paths may be thought of in terms of being either observed (i.e., a series of recorded network locations), or as expected (i.e., determined in an *a priori* manner).

Observed paths may be thought of in terms of whole or partial paths. Whole paths are simply paths between an identified origin and destination. Partial paths may be either a static segment of a whole path or a path that lengthens dynamically over time with or without a predetermined destination. For this effort we focus on the latter, paths that evolve over time with no predetermined destination. While any network space may be used, we express observed paths via serial latitude and longitude locations and, in turn, associate these observations with the nearest network vertices in a planar embedded street network.

Expected paths are determined in an *a priori* manner and represent idealized versions of paths that will be observed. Expected paths are a set of edges, specified via either previously recorded locations, algorithmically via a shortest path between two points, or manually via an appropriate interface).

In order to determine whether an observed path is departing from an expected path two assumptions are necessary. First, for a variety of reasons, an observed path through a network may deviate from what is expected but may still reasonably be considered to be the same (e.g., a parallel road used to divert around an obstruction in a street network). Thus, the basis for determining when an observed path has substantively departed from what is expected must be couched in terms appropriate to the problem at hand.

Allowance for relative path similarity is accomplished through the establishment of a “decay” function around the expected path. This decay function serves to distribute the highly discrete information associated with a specific path on to adjacent edges in an exponentially declining manner relative to the cumulative shortest-path distance to each node encountered in the expected path. We call this representation a “posterior weighted graph” (PWG) and it is the model against which observations are compared.

Identification of departures of observed data from expected paths in real-time also requires the establishment of lower bound criteria for when an observed path is no longer functionally equivalent to an expected path. This lower bound is determined by two parameters, the maximal rate of change of observed data relative to the expected path models, and a threshold time in which no new maxima occur.

In the next section, we formalize the modelling approach. We briefly describe the development of the posterior weighted graph models, the classification process used to compare observations to expectations, and our real-time implementation of this process.

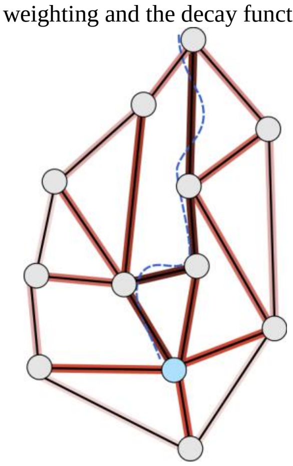
3 Detection of path anomalies

3.1 Characterization of expected paths

As mentioned in the previous section, expected paths are characterized using posterior weight graphs. The PWG probabilistically represents the likelihood that any edge will be used in association with an expected path.

The PWG is created by first initializing every edge in the graph with a 0 weight. The vertex sequence associated with the expected path is then traversed, and the coincident edges are each assigned an initial weight value. Following the assignment of the initial weight value (usually the edge length, but any weight may be used), the edges in the neighbourhood of each vertex are then assigned progressively lower weights using $e^{-dist(i,j)/\sigma^2}$ where *dist(i,j)* is the cumulative shortest-path distance to the next vertex or vertices, and σ is the decay parameter. The depth of the neighbourhood traversal is limited by parameter T_w, a threshold weight tolerance below which the decay is considered to render edge weights negligible in terms of their influence on the model.

Figure 1: Edge weighting and the decay function.



As Figure 1 illustrates, the edges associated with the expected path (dashed line) are most strongly weighted. The edges immediately adjacent are weighted somewhat less strongly, and distant edges are weighted in a very limited manner. The process of traversing vertices in the path is repeated until the expected path is complete. Because the neighbourhood of each vertex is examined, edges coincident to multiple vertices are reinforced.

3.2 Classifying a path

3.2.1 Converting PWGs to a probability model

As there may be multiple PWGs for multiple expected paths, it is necessary to set the stage for modelling any observed path as a set of edge probabilities. This supports a classifier that uses a probabilistic approach to determine from which expected path a set of observed edges would be most likely drawn. The probability model for each expected path is derived from the corresponding PWG.

We begin the process by establishing a minimum edge probability, p_{min}, an arbitrarily low probability that ensures that no edge has zero probability. We then rescale all the edge weights based on the maximum weight less p_{min} and add p_{min} to all of the probabilities (Eqs. 1 and 2).

$$w_{max}^{k} = \frac{max_{w \in W_k} w}{(1.0 - p_{min})} \quad (1)$$

$$p_{ij}^{k} = \frac{w_{ij}^{k}}{w_{max}^{k}} + p_{min} \qquad (2)$$

This scaling process is repeated for each k expected paths and ensures all weighted edges from the PWGs have some minimum probability, that the weight values are monotonically proportional to edge probabilities, and that all potential expected probability models are scaled to the same p_{min}.

3.2.2 The anti-model

Determining when observations depart from expectations as represented by the set of expected probability model requires an additional mechanism. Specifically, as the classifier will identify the probabilistically "best" match even if the corresponding probabilities are very low, we must provision for the case when the path being classified does not strongly match any of the individual expected path probabilities. In order to facilitate this process we develop what we refer to as the "anti-model."

The anti-model is essentially a reciprocal set of probabilities associated with edges not reinforced by the k expected models. First, for each edge in each of the k expected models, the maximum probability for that edge is determined. The anti-model probability is, in turn, calculated for each edge as the minimum of either the complement of the maximum probability or a user-defined parameter, $p_{sensitivity}$ (Eq. 3).

$$p_{i,j}^{anti-model} = min\{1 - p_{i,j}^{max}, p_{sensitivity}\} \quad (3)$$

This results in a final probability model wherein edges with high probabilities in any of the k expected models are assigned a low probability through the $p_{sensitivity}$ parameter. This mechanism implements a heuristic, worst-case identification of an anomalous path. Such a worst-case identification reduces the possibility of falsely identifying anomalous paths. We now explore how this is used in the classification process.

3.2.3 Classifying an observed path

Given the probability models associated with each expected path and the corresponding anti-model, we wish to determine whether an observed path is most like one of the k expected paths or most like the anti-model.

In order to do to this, we compute the log likelihood of the observed path being from any given model (Eq. 4). This represents an assessment of the likelihood of the joint event that the edges in the path set came from model k under the assumption that edge inclusions are conditionally independent given the model.

$$Pr\{path \text{ from model } k\} := \sum_{(i,j)\in path} \log p_{ij}^{k} \quad (4)$$

It is unlikely that the assumption of conditional independence is completely valid. Nevertheless, we believe that the graph contains sufficient information so that proceeding with the naïve assumption still results in a useful classifier.

Finally, for classifying paths, we will typically include the anti-model in addition to the k expected models. After all log likelihoods have been calculated, the model with the highest log likelihood is the model from which the observed path is most likely drawn. If, on the other hand, the anti-model has the highest log likelihood, then we assert that the observed path does not likely match any of the expected paths and can, therefore, be considered anomalous.

3.3 Real-time detection of anomalies

Once the classifier is established, extending it to work with real-time observations is relatively straightforward. Simply, we consider a path to have a starting observation and, over time, successive additional increments of the path are added. In terms of classifying a dynamic set of observations, for each successive observation, the cumulative "observed" path is extended and the classifier is reapplied relative to the original expected models and corresponding anti-model. The challenge is detecting when a set of observations has transitioned from an expected state to an anomalous state.

In order to detect transitions from expected to anomalous in real-time, we use a second order numerical approximation of the backwards difference technique (Eq. 5).

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{\Delta x} \quad (5)$$

For each additional observation (extension to the path), we instrument the real-time classifier to record the log likelihood for the each of the k models and the anti-model. When the trend with the highest log likelihood simultaneously expresses a maximum positive rate of change, we consider this a trigger (indicating the potential for association of the observations with a corresponding model). When the log likelihood for that trend does not decrease for a user specified number of additional "lock in" observations (L_o) and there are no additional triggers, the observed trajectory is considered to be similar to the corresponding model. If this is one of the expected models, then the observed data are considered expected, if the lock-in is associated with the anti-model then the observations are considered anomalous.

3.4 Summary

As with any modelling effort, the success of the model is dependent on the proper selection of the parameters underlying the model. The advantage of having an adequate parameter space, however, is that the model can be tailored to multiple modelling scenarios. For example, while our case studies use spatially embedded transportation networks (and have the commensurate topological constraints), the parameters would allow for use of other networks such as telecommunications networks, social networks, and utility networks. For the scenarios that follow, Table 1 summarizes

Table 1: Model parameters and description.

Parameter & Value	Description
σ =20.0	The rate of decay of edge weights associated with the model for each path. Larger values result in a more general model.
T_w = 0.00001	The weight tolerance controlling the depth of the decay function.
p_{min} =0.0001	Minimum probability for rescaling edge weights from decay model into probability model.
$p_{sensitivity}$ =0.2	A lower bound to limit false positive associations with the anti-model.
L_o = 3	Lock-in. This is the number of post-trigger observations required to confirm association with either an expected model or the anti-model.

the parameter space. In the present experiment, the parameter values were empirically identified and work across a variety of scenarios and input data. Future research will examine how appropriate parameter values can be derived through machine learning based on input training data.

In the next section we illustrate the use of the above model and demonstrate its use for both real-time path matching and anomaly detection.

4 Implementation and evaluation

4.1 Two scenarios

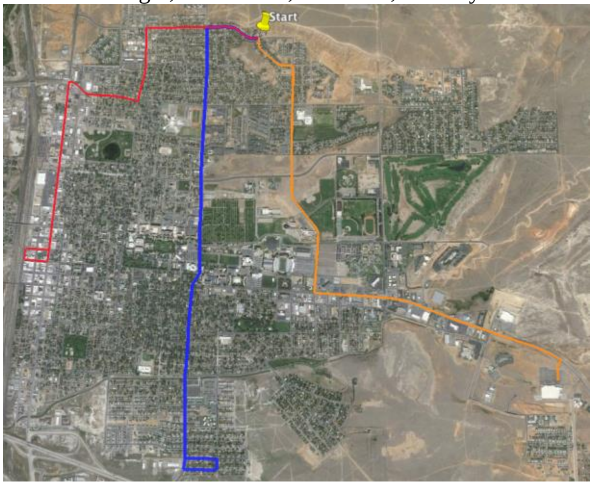
In order to demonstrate the ability of the previously described model to both identify when an observed path matches an expected path and when an observed path becomes anomalous, we present two related scenarios.

- Scenario 1: Multiple expected paths are learned and the anti-model is computed. Observed data are monitored and the point at which the observations are definitively associated with one of the expected paths is reported.

- Scenario 2: Multiple expected paths are learned and the anti-model is computed. Observed data are monitored and the point at which the observations can definitively be considered anomalous is reported.

The scenarios are based on a subset of the street network from Laramie, WY, USA with expected data based on routes derived from Google Maps and observed data collected using a Garmin Forerunner 210 GPS watch.

Figure 2: The learned paths shown on the Laramie streets. From left to right, Downtown, Southend, Grocery.



Source: Google Earth.

Both of the scenarios classify against three learned models and the anti-model. The three models include “Downtown,” round trip travel to the Laramie town centre, “Southend,” an arbitrary trip across town and, “Grocery,” a trip to the grocery store. All of the paths were mapped in Google Maps, extracted as GPX data, and mapped to coincident vertices in the street network using a spatial search algorithm. The resultant ordered vertices are the graph-based representation of the potential expected paths.

Though the observed data were collected as a single GPS track, we simulate real-time online processing. The real-time emulation is accomplished by introducing each successive track point and extending the observed path. We then re-compute the log likelihoods, recalculate the numerical approximation of the second derivative, and evaluate against L_o.

4.2 Scenario 1 – Multiple expected paths

In this scenario we simulate where an individual is choosing from among several potential activities as specified in Section 4.1. Our goal is to observe their trajectory and, as quickly as possible, identify which activity they are most likely doing.

As Figure 3 illustrates (and can be intuited from Figure 2), it is not possible to differentiate the activity based on the initial set of observations. However, beginning with the fourth observation (49 seconds into the journey), the likelihood of any given path begins to diverge. The first trigger (maximum positive rate of change associated with the highest log likelihood) occurs at the 6th observation (81 seconds), and the association with the Southend route is locked in at the 9th observation, approximately 51 seconds later.

Figure 3: Confirmation of the Southend path at 132 seconds.

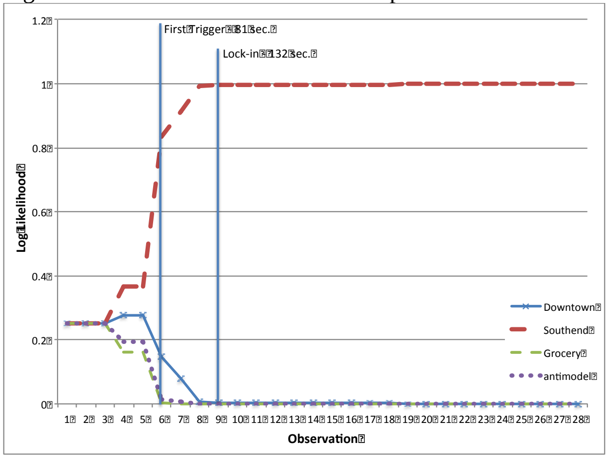
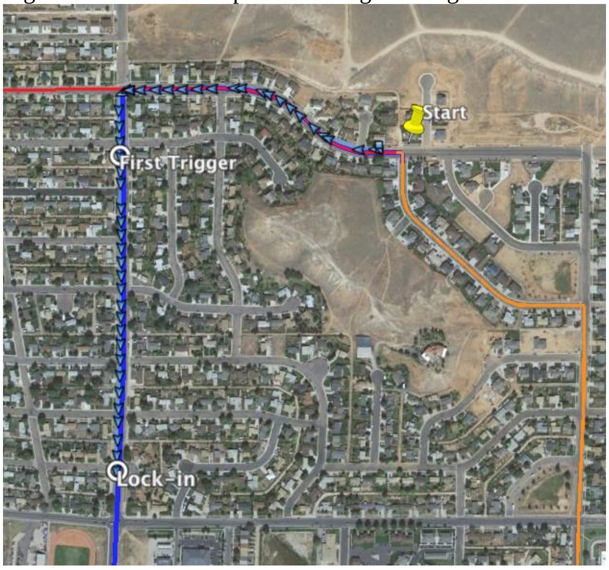


Figure 4 shows a map view of the first trigger and the subsequent lock-in. The “soft” association that comes from the trigger event helps minimize false positives and serves to leverage the fuzziness (and the potential that observations may match, depart, then return to a specific expected path) facilitated with the underlying decay function described in Section 3.1.

Figure 4: The observed path from origin through lock-in.

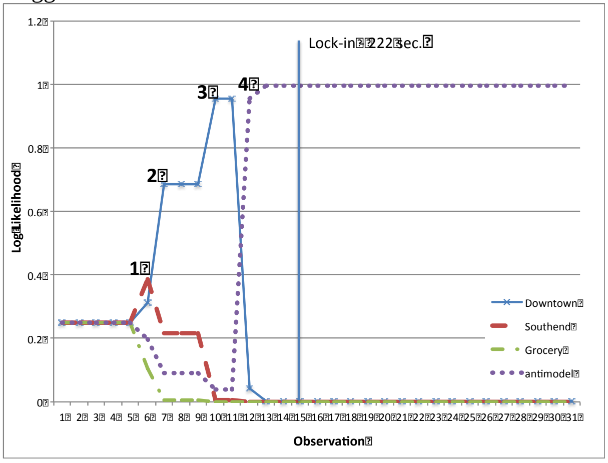


Source: Google Earth.

4.3 Scenario 2 – Expected vs. anomalous paths

Like the previous example, here we simulate a scenario wherein we are trying to determine if an individual's trajectory through the network is consistent with one of three predetermined paths. In contrast, however, rather then reporting when an individual's trajectory is associated with an expected path, we want to report when their trajectory is definitively anomalous.

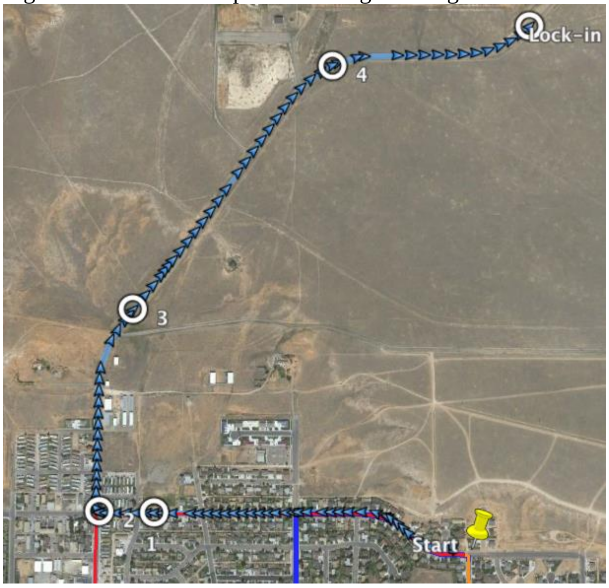
Figure 5: Confirmation of anomalous route at 222 seconds. Trigger events are shown with numbers 1 – 4.



In contrast to the previous example, the first trigger event in this case arises from an apparent association with the Southend path at 79 seconds (Figure 5, trigger 1). As illustrated, however, this is something of a false positive, and the lock-in fails with a second trigger event at 96 seconds in association with the Downtown path. This association is relatively strong, however, a third trigger event on the same model occurs with the 10th observation at 142 seconds. This third trigger event prevents the lock-in that would have otherwise occurred at this observation. At the 12th observation (174 seconds) a forth trigger event, this time associated with the anti-model, is seen. Three observations later (per T_w), there are no additional triggers and the lock-in as an anomalous path is confirmed at 222 seconds.

Again, Figure 6 shows a map view of the observed trajectory and the various detection events.

Figure 6: The observed path from origin through lock-in.



Source: Google Earth.

The sequence of triggers illustrates the role of the interacting decay functions in terms of defining the probabilities of associating with any given path. Since the probability of the observed data is cumulative in nature, there is a seeming lag between the path association and the trigger point. This is a characteristic of the approach and can be adjusted through the sensitivity and σ parameters.

4.4 Summary

In the presented scenarios, the paths themselves and the corresponding GPS data can clearly be differentiated from one another and the underlying anti-model using the presented method and corresponding parameters.

In a different context (e.g., that such as illustrated in [3]), the same approach could be used to determine whether a single GPS track is more like a collection of potential paths or, again, the anti-model. A characteristic of this approach is its flexibility supporting either comparisons to specific, individual paths or, alternatively, a collection of paths traversing the network in question. The learned anti-model can be the "reciprocal" of a single path or a collection of paths or segments. The application in question will be the key driver in decisions regarding the overall representation, definition of path start and end points, and whether or not specific, individual paths or path sections need to be identified.

5 Conclusion

This paper presents a preliminary method for using a classification-based approach for real-time interpretation of network observations. The presented approach is useful for discerning either when a set of observations is most similar to an expected path or unlike any *a priori* specified expectations. This latter case is useful for identifying anomalous paths in real-time.

The ability to detect either path similarity or difference is predicated on learning the model or models that characterize expected data. These models, along with the anti-model must be learned in the context of the specific problem at hand, the nature of the corresponding network, and the characteristics of the observed data. The parameters, while perhaps numerous, allow for the approach to be tailored to a variety of scenarios. While the presented approach is on a street network, any network with the potential for supporting expected and observed paths is a candidate for use with this method as the entire process is aspatial and based on network measurements and network locations.

Two key areas merit additional research. First, as previously mentioned, it would be useful to be able to learn the parameter space for different problem classes. This would enable more effective parameter selection depending on problem and network characteristics. The second area for additional research is in terms of improving the approach for handing real-time data. Predictive methods from the signal process and machine learning communities may prove very useful in this regard.

References

[1] J. A. Alvarez-Garcia, J. A. Ortega, L. Gonzalez-Abril, and F. Velasco. Trip destination prediction based on past GPS log using a hidden markov model. Expert Systems with Applications, 37(12):8166–8171, 2010.

[2] S. Buthpitiya, Y. Zhang, A. K. Dey, and M. Griss. N-gram geo-trace modeling. In Proceedings of the 9th International Conference on Pervasive Computing, Pervasive’11, pages 97–114, Berlin, Heidelberg, 2011. Springer-Verlag.

[3] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, and S. Li. Real-time detection of anomalous taxi trajectories from GPS traces. In Mobile and Ubiquitous Systems: Computing, Networking, and Services, pages 63–74. Springer, 2012.

[4] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data, SIGMOD ’05, pages 491–502, New York, NY, USA, 2005. ACM.

[5] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In Image Processing, 2005. ICIP 2005. IEEE International Conference on, volume 2, pages II–602. IEEE, 2005.

[6] Q. Lu, F. Chen, and K. Hancock. On path anomaly detection in a large transportation network. Computers, Environment and Urban Systems, 33(6):448 – 462, 2009.

[7] T.-S. Ma. Real-time anomaly detection for traveling individuals. In Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, Assets ’09, pages 273–274, New York, NY, USA, 2009. ACM.

[8] F. Porikli. Trajectory distance metric using hidden markov model based representation. In IEEE European Conference on Computer Vision, PETS Workshop, volume 3, 2004.

[9] A. A. Sodemann, M. P. Ross, and B. J. Borghetti. A review of anomaly detection in automated surveillance. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 42(6):1257–1272, 2012.