Entity Reconciliation in a Multi-camera Network

Raghu Ganti
IBM T J Watson Research Center
Yorktown Heights, NY
rganti@us.ibm.com

Mudhakar Srivatsa*
IBM T J Watson Research Center
Yorktown Heights, NY
msrivats@us.ibm.com

B. S. Manjunath
University of California, Santa Barbara
Santa Barbara, CA
manj@ece.ucsb.edu

ABSTRACT

Location traces are becoming fairly abundant with the introduction of various mobile devices such as smartphones, in-car navigation units, and video cameras. Each individual type of device generates different features about a mobile entity along with the location of that entity itself. For example, the smartphone can provide the motion (using accelerometer) of an individual, whereas a video camera can identify what type of clothing the person is wearing. A key challenge is to be able to fuse the data across different data sources and generate a unique view for each entity. This paper tackles a slice of this larger problem, which is to *reconcile* entities across a multi-camera network and a GPS trace from a smartphone and proposes a novel algorithm that can scale horizontally to adapt to new age distributed systems such as Apache Spark and IBM's InfoSphere Streams. We show through extensive experiments on a real-world dataset that our algorithm outperforms existing approaches and adapts to horizontally scalable distributed environments.

CCS Concepts

•Information systems → Data analytics; •Computing methodologies → Distributed algorithms;

Keywords

Track reconciliation; Distributed spatiotemporal analytics

1. INTRODUCTION

Various types of sensors that sense the environment around us are increasingly becoming common and the data from these are readily available for consumption. Some of the examples of such sensed data is call-detail records from telecommunication companies, GPS data from various social media apps (e.g., Twitter, Facebook, Google maps) and in-car navigation units, and video data from camera networks. Consumption of this data for *single-track* applications is already quite prevalent, for example, traffic analysis from location data generated by mobile devices, crowd identification from call-detail records [8], and security and surveillance from multi-camera networks [22]. However, data fusion across these various sources is a problem that has received inadequate attention in the past. This paper tackles the problem of data fusion across multiple sources of sensors, specifically across a multi-camera network and a telecommunication company's call-detail records. In particular, we address the problem of being able to fuse these data sources using location traces identified in the multi-camera network with that of those available from mobile phones.

A simple approach to this problem, which is based on past work [17, 15, 23], would be to reconcile the entities across multiple views (or cameras) using unique features of the entities. For example, spatiotemporal patterns are used in [17] to reconcile entities across multiple cameras, whereas colors are used in [15]. The basis of such an approach is to do *matching* across multiple cameras. Our focus is on location in this paper and we will use the well known mix-zone based entity matching across multiple cameras. However, the lack of fusing other sources of data, such as the mobile phone location traces reduces the effectiveness of this approach (which we show later gives accuracies of $< 25\%$).

Data fusion approaches are usually more effective in this regard as they take into consideration diverse data sources and combine them based on commonalities. In this paper, we use location traces as the common measure to combine multi-camera video feeds with that of mobile phone records. The fusion problem can be framed as follows - consider that the multi-camera feed generates *tracklets* for each camera view and our goal is to combine these tracklets from multiple views based on a single track generated by the mobile phone records. This problem can be formulated as classical alignment of two time-series such that a cost metric between these time-series is minimized. This is a classical Dynamic Time Warping problem [13], which has a slow optimal solution and fast sub-optimal variants (Itakuara parallelogram and Sakoie-Chiba). We show how these algorithms can be applied for fusing data across multiple sources in the context of our problem. Finally, we develop a novel technique that has its roots in computing what are called as space-time boxes and generating *track signatures* based on these space-time boxes to fuse data across multiple cameras and mobile phone records. The key novelty of this approach is its ability to horizontally scale to new age distributed systems such as Apache Spark [1] and InfoSphere Streams [10], where the location and timestamp data are *hashed* to main-

*First and second authors equally contributed to this work

ICDCN '16, January 04-07, 2016, Singapore, Singapore

DOI: http://dx.doi.org/10.1145/2833312.2849566

tain key properties for achieving scalability, which include (i) Determinism, (ii) Extensiblity, (iii) Uniform density, and (iv) Fast. Extensive experiments performed on a real-world dataset shows that the STB technique is much faster (2-6x) than the existing approaches while being as accurate as these approaches.

The rest of this paper is divided into five sections, Section 2 describes the data characteristics used in this paper, the problem is formulated and different solutions are presented in Section 3, the algorithms are evaluated from a performance and accuracy standpoint in Section 4, related work is presented in Section 5, and finally, we present the conclusions and future work in Section 6.

2. DATA CHARACTERISTICS

We will explain the characteristics of the data that will be used in the rest of this paper. Our base mobility traces are obtained from a telecommunications company operating in a densely populated region in an Asian country. The dataset consists of Call Detail Records (popularly known as CDRs) from about 10 million unique users over a period of one week. These CDRs are generated by a voice-call or an SMS and include information such as IMEI, cell tower location associated with the call, and the duration of the call.

We will provide some brief characteristics of mobility pertaining to this dataset. We consider one day's worth of data and plot the histogram of number of samples for each individual, this will give us a basic understanding of how often location samples are obtained from an individual. We illustrate this histogram in Figure 1. We observe from this figure that the number of samples per individual for the CDR dataset for about 30% of the population is around 20 samples/day. This observation is in line with typical mobile phone usage, where calls and SMSes require manual involvement.

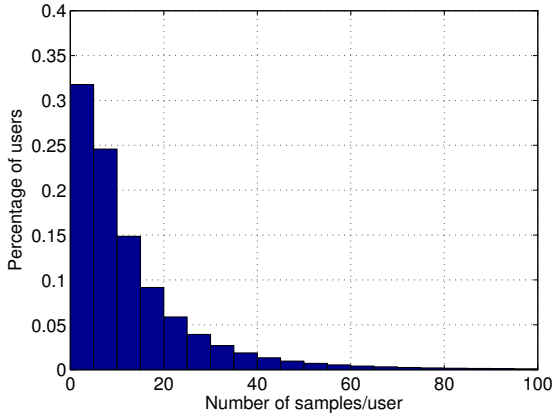


Figure 1: Number of samples per user for CDR dataset

We summarize the data characteristics in Table 1.

Characteristic	CDR
Number of users	10m
Time duration	1 week
Data types	Call/SMS
Median samples/day	8

Table 1: Summary of characteristics of the CDR dataset

3. ALGORITHMS

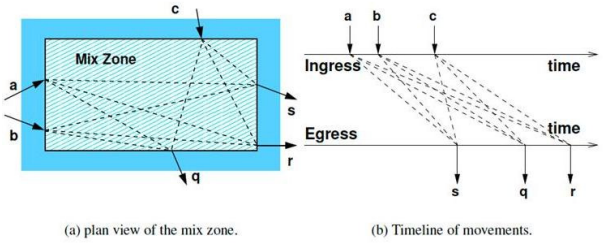
In this section, we present three different types of algorithms for reconciling entities across a multi-camera network. The problem formulation across the three different algorithms is the same, i.e. to check if two entities' trajectories match. First, we assume that there are two sources of location information - one is from a video camera and the other is from a GPS device (e.g., the GPS on the mobile phone, call-detail records generated by telecommunication companies). On a single camera, the identifier of a single entity is constant and is assumed to be identifiable - i.e., individual entities can be differentiated across the time-period in which the entities dwell in the given camera's view. However, this entity identity is lost across multiple cameras. Our aim is to correlate the location sequence generated by the GPS device with that of the trajectories generated by the multiple cameras and reconcile the identities of the entities across these multiple cameras.

The first approach is to use only features of the objects and match them across different cameras, this is based on the concept of mix zones and relies on weighted bipartite matching. The second is to compute an optimal alignment between trajectories generated by the GPS device and those generated by the cameras using Dynamic Time Warping. Finally, the main contribution of this paper is to apply the notion of space-time boxes to generate track signatures and compare these across the trajectories from the cameras to that of the GPS devices. We will explain each of these approaches in further detail and show that the space-time box approach allows for distribution across a cluster of nodes and hence is amenable to horizontal scaling when compared to other approaches.

3.1 Mix Zones

The concept of a mix [4] was introduced by Chaum in 1981. Since then several authors have used *mixzone* as a network routing element to construct secure networks such as Onion routing [9] and Tor [6]. In recent years, the same technique was used in the context of location privacy [16, 11]. Figure 2 illustrates the notion of a location based mixzone. The figure shows a mixzone (typically a region) wherein three entities $\{a, b, c\}$ enter the mixzone and three entities $\{q, r, s\}$ exit the mixzone. In the context of entity reconciliation across multiple cameras, the idea is to determine which entities are linked to each other. However, an ideal mixzone erases all mapping information between the entities $\{a, b, c\}$ and $\{q, r, s\}$. Indeed, a third party that has access to only the mixed data would only know that three entities entered a mixzone and three exited the mixzone − however, any of the $3! = 6$ mappings between the ingress and egress pseudonyms is plausible. One possibility to address this issue is to keep track of features of each entity and try to associate these features based on some distance metric. For example, camera A that captures the locations of entities in mixzone 1 can use the features of the entities to relate entities captured by camera B. Typical ways to tackle such entity reconciliation problem is to use a weighted bipartite

matching algorithm, which will be our first approach.

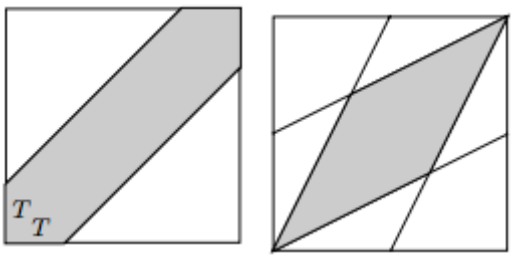


(a) plan view of the mix zone. (b) Timeline of movements.

Figure 2: Mixzones in Location-based Regions

3.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm that measures similarity between two temporal sequences which may vary in time or speed. Consider two sequences $X = (x_1, x_2, \ldots, x_N)$ and $Y = (y_1, y_2, \ldots, y - M)$ of lengths N and M, respectively. Let us fix a feature space F, then $x_n, y_m \subset F$ for $n \subset [1 : N]$ and $m \subset [1 : M]$ and a *local cost measure (or local distance measure)*, $c(x, y)$. $c(.)$ is low if x and y are similar to each other and high if they are different from each other. The goal of a DTW algorithm is to find an alignment between X and Y which has minimal overall cost. Intuitively, such an optimal alignment runs along the "valley" of low cost within the cost matrix defined by $c(.)$. Computing the optimal DTW requires $O(NM)$, which is computationally expensive. In our case, we need to find the optimal alignment between traces generated across multiple cameras and the traces from other location data sources (e.g., GPS, mobile call detail records). A common technique to speed up DTW computation is to impose global constraint conditions on the admissible warping paths. Mathematically, if $R \in [1 : N]X[1 : M]$ is a subset referred to as global constraint region, then a warping path relative to R is a warping path that entirely runs within the region R. The optimal warping path is relative to R. Two well-known global constraint regions are the *Sakoe-Chiba band* and the *Itakura parallelogram*, which are shown in Figure 3. More information on DTW and the different global constraint regions can be found in [13].



(a) Sakoe-Chiba band (b) Itakura parallelogram

Figure 3: Illustration of global constraint regions

3.3 Space Time Boxes

Our final approach and the main contribution of this paper is to use the notion of space-time boxes (STB). We note that a significant drawback of the above approaches – Mix zones (Section 3.1) and DTW (Section 3.2) is that the operations are not easy to parallelize to accommodate for new age distributed systems such as Spark [1] and InfoSphere Streams [10]. In order to discretize the problem and enable parallelizable operations, we introduce the notion of a space-time box, which discretizes a location and timestamp into a 2-D box and a time range. In the case of a GPS generated location in latitude/longitude, this 2-D box is a *bounding box*. Our approach involves the generation of a *hash* that has certain key properties (enabling uniform scaling in a distributed environment):

Deterministic hashing: An object's location and timestamp are deterministically mapped to a small set of keys. In general, each hash value h covers a region such that all points within that region are mapped to the same value h. This yields keys that can be directly used in a distributed environment for horizontal scaling

Extensible/telescopic hashing: An object's location and timestamp are mapped to an extensible key such that mapping at different spatial resolutions result in consistent key assignment. An example of extensible hashing over two dimensional coordinates with gradual precision loss is as follows:

$$\begin{aligned} hash(40.00105, -78.30105) &= dr07d1yzj21 \quad (1) \\ hash(40.001, -78.301) &= dr07d1yy \\ hash(40.01, -78.2) &= dr07se \\ hash(40, -78) &= dr0e \end{aligned}$$

Uniform density: The hash technique must support a choice of keys such that given any set of points, the number of points mapped to a given key is nearly equal for all of the keys. This requirement must hold independent of the distribution of the points in the spatial domain, i.e. even if the points occur in clusters they need to be mapped uniformly to keys. This property is essential to avoid "hot spotting" in distributed systems wherein a node in a compute cluster may get overloaded because a disproportionately large number of points are mapped to that node. Avoiding hotspots is essential to get linear scalability with respect to compute cluster size.

Bit Arithmetic: This enables for manipulation of keys and performing various operations on these keys such as truncate, distance between keys, and identifying neighbors in the 3D space in an extremely fast manner.

The STB is realized as a combination of a geohash [14] and a time range, where the location in latitude/longitude is first mapped to coordinates in $[-1, +1]$ range, followed by interleaving the bit representation of each of the mappings to generate an interleaved bit representation of the mapped location. The discretization depends on the granularity of the space box, the coarser the granularity, the larger the space box. Two entities that map to the same space box will be within the specified distance.

Track Signatures and Reconciliation

Now, that we have described how to generate space-time boxes given a location and timestamp (which is the fundamental representation of a trajectory), we will explain how the entity reconciliation works. We generate *track signatures* given a sequence of locations and timestamps. This track signature is generated by concatenating the STBs across se-

quential locations (and their corresponding time stamps). The algorithm to *reconcile* two tracks is then rather simple, it is done by checking if two entities have the same signature (i.e., same hash across a sequence of timestamps).

4. EVALUATION

We will evaluate the algorithms presented in the previous section from a run-time performance and accuracy standpoints. The algorithms that will be evaluated are (i) Mix-zone weighted bipartite matching, (ii) STB track signature, (iii) Itakura parallelogram, (iv) Sakoe-Chiba, and (v) optimal DTW. The mix-zone based algorithm is a baseline as it does not do fusion across multiple data sources, whereas the others fuse multiple data sources. Since, it is extremely difficult to obtain real data for evaluating these algorithms, we generate fake traces from the data that was described in Section 2 as follows. We consider a large area that covers a populated portion of the city from which the data was collected and select about 50 entities. For each entity, we chose the longest track in that region. We perturb each entity's track and create *tracklets* that signify the location traces for each view of a multi-camera network. Further, we assume that location of entities from video frames can be extracted, which has been addressed in the past [5].

First, we measure the run-time of each of the algorithms to reconcile entities as the number of entities are varied (all our experiments are performed on a x86 server with 64GB RAM and no other workloads) and plot this in Figure 4. We observe from this figure, which has y-axis as log scale to show the DTW run-time, that the fastest execution time is that of mix-zones and the slowest is that of DTW. The STB approach is significantly faster than the Itakura parallelogram or Sakoe-Chiba approximations to DTW, which can be as much as 6x faster when the number of entities considered are around 40. We chose a track length of 3 and a band size of 3 for the algorithms (ii), (iii), and (iv).

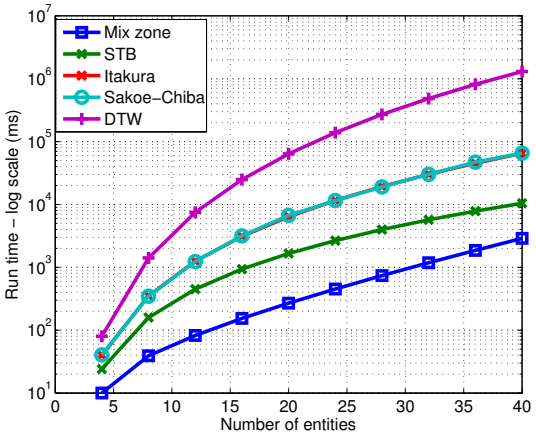


Figure 4: Run-time performance as number of entities are varied

Next, we plot the accuracy, as a fraction of the number of entities reconciled correctly, as the number of entities are changed in Figure 5. A key observation is that even though mix-zone algorithm was much faster than all the other techniques (Figure 4), the accuracy is very poor (it reduces to $< 10\%$ when the number of entities are 25 or greater). Whereas, DTW optimal approach has the best accuracy, which is to be expected, as the other approaches are approximations. We can consider the STB track signature approach as a DTW approximation in that the global constraints are flexibly chosen based on the granularity. Although, we note that the accuracy loss due to these approximations is not significant even when the number of entities are increased.

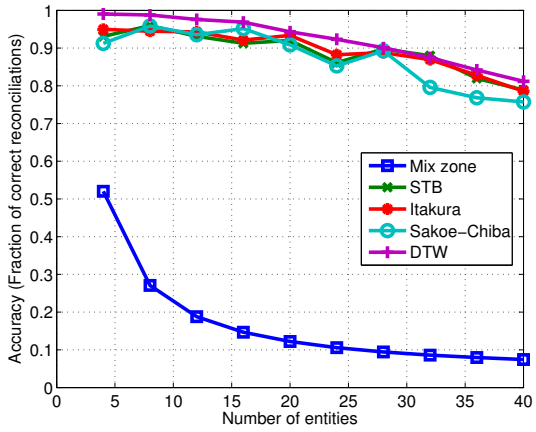


Figure 5: Accuracy - fraction of correct reconciliations as number of entities are varied

We modify the amount of noise added to the tracklets and plot the accuracies for each algorithm in Figure 6. We observe that the mix-zone based algorithm has very poor accuracy as expected, with the other approaches gradually decreasing in their accuracies as the noise added is increased. We note that with noise as high as 800m, the accuracy loss is about 20%. This suggests that the camera parameters (e.g., field-of-view, features extracted) should be chosen in such a way that the noise is within a given error margin.

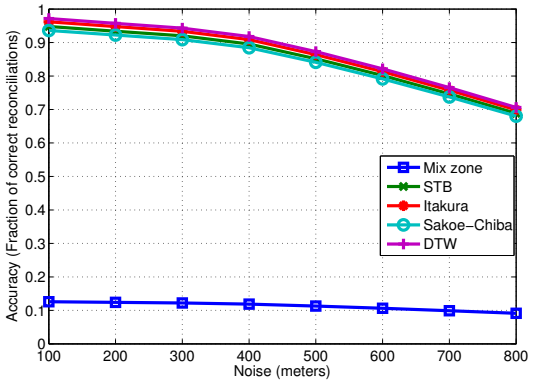


Figure 6: Accuracy - fraction of correct reconciliations as noise added is varied

Finally, we plot the accuracies, when the track length (for the STB algorithm) or the band size (for Itakura parallelogram and Sakoe-Chiba) are varied, in Figure 7. Since this is not applicable to the mix-zone and DTW algorithms,

we will not show them in this Figure. We observe from this that there is a significant jump in terms of accuracy when the track length (or band size) is changed from 1 to 3, whereas further increases do not effect the accuracy significantly. This suggests that a track length (or band size) of about three should be sufficient for our purposes.

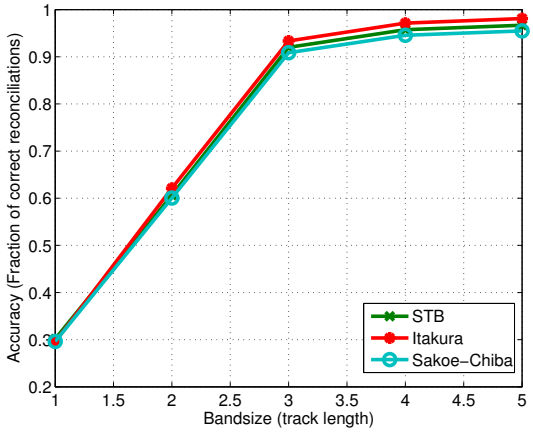


Figure 7: Accuracy - fraction of correct reconciliations as track length (band size) is varied

We observe from the above experiments that the best algorithm in terms of accuracy is DTW, but it takes a long time to run. Our proposed approach is comparable in terms of accuracy to that of the Itakura parallelogram and Sakoe-Chiba algorithms, but outperforms them in terms of runtime significantly. Moreover, our approach is amenable to horizontal scaling, thus making the overall solution rather attractive.

5. RELATED WORK

We divide the related work into two parts, one that uses only video for object tracking and the other that uses other forms of sensor data for object tracking. In the computer vision literature, object detection is typically defined as detecting instances of semantic objects of a given class (e.g., humans, buildings, and vehicles). On the other hand, object recognition extends the object detection to identify the particular object in an image or video sequence. For example, identifying the person when a human is detected and identifying the building once it has been detected. Several algorithms have been developed for object detection [21, 7, 20] as well as object recognition [12, 3, 19]. These algorithms are typically based on machine learning techniques combined with image feature extraction. Our work differs from these techniques in that we consider a distributed multi-camera network and apply fusion based techniques that combines location data from multiple sources to achieve object tracking.

Object tracking work has been extended to multi-camera settings in [17, 15, 23], where the problem being addressed is that of tracking a given object across multiple cameras. These approaches focus on fusing data from multiple cameras and identifying and tracking the object of interest across these cameras. However, our approach is to combine non-camera data sources with that of the camera to achieve object tracking.

Tracking objects using other forms of sensor data has also been explored in the past. For example, in [2], locations of individuals inside a building were tracked using RF-signal strength measurements. In a wireless sensor network, mobile agents combined with sensor data from the deployed sensor network are used to achieve location tracking [18]. Such location tracking techniques primarily rely on fusing sensor data from distributed sensor networks, however there have been no efforts in fusing images/videos with location data from mobile devices. In this paper, we explore this possibility and introduce the notion of combining location from a distributed camera network with that from mobile devices to achieve object tracking.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a novel solution based on Space-time boxes to reconcile entities in a multi-camera network. We formulated the problem of fusing two diverse sources of location data - a multi-camera network and mobile phone records - as a time-series alignment problem. We show that the existing optimal solution is too slow for practical use. Further, we show that the STB solution is much faster than the sub-optimal solutions for DTW, the Itakura parallelogram and Sakoe-Chiba algorithms. Also, we show that the STB solution is amenable for horizontal scaling on new age distributed system platforms such as Apache Spark and InfoSphere Streams. We evaluated these algorithms on a real-world dataset and showed that STB approach is 2-6x faster than the state-of-the-art algorithms while preserving the accuracy of reconciliation. As future work, we will explore algorithms for scaling out and improving video feature extraction based on the reconciled entities.

7. ACKNOWLEDGMENTS

We would like to thank the telecommunication companies anonymously for providing the datasets for analysis. Research reported in this paper was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

8. REFERENCES

[1] Apache Spark. Apache Spark. http://spark.apache.org/.

[2] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proc. of Infocom*, volume 2, pages 775 – 784, 2000.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509 – 522, April 2002.

[4] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. In *Communications of ACM, 24(2): 84-88*, 1981.

[5] D. L. de Ipina, P. Mendonca, and A. Hopper. Trip: A low-cost vision-based location system for uniquitous computing. *Personal and Uniquitous Computing*, 6(3):206 – 219, May 2002.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second generation onion router. In *13th USENIX Security Symposium*, 2000.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627 – 1645, September 2009.

[8] R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, November 2011.

[9] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. In *Communications of ACM, Vol 42(2)*, 1999.

[10] IBM Infosphere Streams. IBM Infosphere Streams. http://www-01.ibm.com/software/data/infosphere/streams/.

[11] X. Liu, H. Zhao, M. Pan, H. Yue, X. Liu, and Y. Fang. Traffic-aware multiple mix zone placement for protecting location privacy. In *Infocom*, 2012.

[12] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of Computer Vision*, volume 2, pages 1150 – 1157, 1999.

[13] M. Muller. *Information Retrieval for Music and Motion*. Springer-Verlag Berlin Heidelberg, 2007.

[14] G. Niemeyer. Geohash. http://en.wikipedia.org/wiki/Geohash.

[15] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. V. Gool. Color-based object tracking in multi-camera environments. In *Proc. of DAGM Symposium*, pages 591 – 599, 2003.

[16] B. Palaniswamy and L. Liu. Mobimix: Protecting location privacy with mix-zones over road networks. In *ICDE*, 2011.

[17] A. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa. Object detection, tracking, and recognition for multiple smart cameras. *Proceedings of the IEEE*, 96(10):1606 – 1624, October 2008.

[18] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. *The Computer Journal*, 47:448 – 460, November 2003.

[19] J. R. R. Uijilings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *Springer Journal of Computer Vision*, 104(2):154 – 171, September 2013.

[20] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proc. of IEEE Computer Vision*, pages 606 – 613, 2009.

[21] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of Computer Vision and Pattern Recogniton*, volume 1, pages I–511 – I–518, 2001.

[22] G. Wu, Y. Wu, L. Jiao, Y.-F. Wang, and E. Chang. Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance. In *Proc. of Multimedia*, pages 528 – 538, 2003.

[23] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney. Real-time wide area multi-camera stereo tracking. In *Proc. of Computer Vision and Pattern Recognition*, volume 1, pages 976 – 983, 2005.