A Two-Layer Approach for Energy Efficiency in Mobile Location Sensing Applications

Yi-Yin Chang, Cheng-Yu Lin, Ling-Jyh Chen

A Two-Layer Approach for Energy Efficiency in Mobile Location Sensing Applications[*]

Yi-Yin Chang, Cheng-Yu Lin, and Ling-Jyh Chen

Academia Sinica, Taiwan
{yyc, ntuaha, cclljj}@iis.sinica.edu.tw

Abstract. Mobile location sensing applications (MLSAs) represent an emerging genre of applications that exploit Global Positioning System (GPS) technology and facilitate location-based services. The design of MLSAs must incorporate a tradeoff between information accuracy and energy efficiency because GPS technology and data communications are energy expensive and unaffordable for most MLSA platforms, which are battery-powered and therefore resource-constrained. In this study, we propose a two-layer solution to address the above issue. The solution comprises an Event-based GPS Tracking (EBT) scheme and an Inter-Frame Coding (IFC) scheme for energy efficiency on the MLSA GPS tracking and data communication layers respectively. Using a comprehensive set of evaluations, as well as realistic MLSA traces, we evaluate the proposed solution in terms of power saving, location accuracy, and the data compression ratio. The results demonstrate that our solution can achieve better energy efficiency while preserving good location accuracy in all test cases. Moreover, the scheme is simple, effective, and generalizable to other mobile location sensing applications.

Keywords: energy efficiency, GPS tracking, location sensing

1 Introduction

With the advances in wireless communications and GPS technology, mobile location sensing applications (MLSAs) are rapidly permeating every part of our living environments. Unlike conventional Internet and mobile applications, MLSAs exploit Global Positioning System (GPS) technology and facilitate the provision of emerging location-based services (LBS). This new genre has already spawned a wide range of applications, such as location-based social networks [1, 2], mobile urban sensing systems [3], and moving object tracking systems [4, 5].

The use of GPS technology in MLSAs raises a new technical and 'green' challenge, namely, excessive power consumption due to *GPS localization* and *stream data communications*. There are two reasons for this phenomenon. First, an MLSA requires timely and accurate location information, so it favors keeping GPS receivers in the ON mode continuously. However, GPS receivers are known to be power-hungry; thus, keeping them "always-on" is unaffordable for MLSAs because the lifespan of the latter is

constrained by the battery capacity of the associated mobile device. Second, an MLSA generates a substantial amount of continuous data, including the trajectory of the moving object (i.e., the location information as a function of time [6]) and other contextual information (e.g., sensory data and event logs); thus, data communications are costly in monetary terms as well as power consumption.

Several approaches have been proposed to solve the above issues. Generally, existing solutions to the first problem (i.e., reducing the energy required for MLSA GPS tracking) can be divided into two categories: *static duty-cycle* (SDC) approaches (e.g., [3, 5]), and *dynamic duty-cycle* (DDC) approaches (e.g., [7–15]). The former turn GPS receivers ON and OFF at regular intervals, while the latter adjust GPS duty cycles based on analytical models or events triggered by additional sensors. Unfortunately, these approaches are not feasible for MLSAs because 1) SDC approaches operate in a "blind" manner, without considering the context of moving objects; and 2) DDC approaches require a lookup table of pre-learned radio patterns, which is not practical for generic MLSA scenarios.

Solutions to the second problem (i.e., reducing the energy required for MLSA data communications) are either byte-level compression approaches (e.g., [16]) or spatio-temporal compression approaches (e.g., [17–19]). These approaches are infeasible for MLSAs because 1) byte-level compression methods compress MLSA data without considering its intrinsic properties, so the compressed data cannot be processed directly without decompression into its raw form; and 2) spatio-temporal compression methods are based on the lossy concept, i.e., they achieve a good compression ratio at the expense of information loss.

Clearly, a comprehensive solution that can address the challenge without loss of functionality for MLSAs is highly desirable. An ideal solution should meet three criteria: 1) it should be self-adaptive with minimal prerequisites for extra knowledge (e.g., pre-learned radio patterns) and sensors; 2) it should provide lossless compression; and 3) it should be simple, lightweight, and portable to off-the-shelf phones.

In this study, we introduce a two-layer approach to improve the energy efficiency and preserve the location accuracy of MLSAs. Specifically, we propose an event-based GPS tracking approach that operates on the GPS tracking layer and triggers a GPS receiver to lock when either *the receiver has been in the OFF mode longer than a threshold* or *a turn event is detected* by the built-in 3-axis accelerometer (i.e., G-sensor) of the latest smart phones. We also propose an *Inter-Fame Coding* (IFC) algorithm that exploits spatio-temporal localities on the data communication layer to compress trajectory-based MLSA data. As a result, the compressed data can be used for spatio-temporal operations in MLSAs without decompression. The algorithm can also be implemented easily in conjunction with unequal erasure protection (UEP) and data prioritization schemes for operations in lossy or resource-constrained environments.

Using realistic bus trajectory data collected from the TPE-CMS system [20], we evaluate the proposed approach in terms of location accuracy, power saving, and the compression ratio. The results demonstrate that the solution can achieve a balance between location accuracy and power saving in GPS tracking, and the IFC approach yields a good compression ratio of about 50%. Finally, the proposed approach is simple, effective, and ready for immediate real-world deployment.

The remainder of this paper is organized as follows. Section 2 contains a review of existing GPS tracking and trajectory compression schemes for MLSAs. In Section 3, we present the proposed Event-based GPS Tracking (EBT) approach for MLSA GPS tracking; and in Section 4, we describe the proposed Inter-Frame Coding (IFC) approach for MLSA data compression. We discuss the evaluation results for the EBT and IFC schemes in Sections 5, and then summarize our conclusions in Section 6.

2 Related Work

One of the most challenging aspects of GPS technology is its power consumption, which is expensive and renders it unaffordable for emerging mobile and battery-powered platforms, such as smart phones and wireless sensor networks. To prolong the lifespan of built-in GPS systems, several *static duty-cycle* (SDC) schemes reduce the power consumption of GPSs by turning them ON and OFF periodically. For instance, the BikeNet [3] and ZebraNet [5] projects use SDC-based approaches and turn their GPS receivers ON and OFF according to a fixed duty cycle. BikeNet turns the GPS receivers ON for one minute every 8 minutes, while ZebraNet turns them ON for 150 seconds every 1,000 seconds. In contrast, *dynamic duty-cycle* (DDC) approaches adjust the GPS duty cycle based on sophisticated models [8, 12, 13] or events triggered by add-on sensors. For example, they adjust duty cycles according to whether users are in outdoor/indoor environments by using radio signal recognition [9], or whether the tracked objects have traveled a certain distance by using accelerometers, gyros, and digital compasses [7, 10, 11, 14, 15]).

On the other hand, to improve the scalability of moving object databases, several trajectory compression algorithms exploit the spatial and temporal localities in trajectory data [17–19]. For instance, Meratnia et al. proposed the *time ratio algorithm* [18], which discards a data point on a trajectory if that point can be interpolated/extrapolated by any two adjacent data points on the trajectory within an error threshold. Gudmundsson et al. [17] developed a fast implementation of the well-known *Douglas-Peucker* line simplification algorithm [21], and reduced the computational complexity from $O(n^2)$ to $O(nlog^k n)$. Finally, Potamias et al. [19] proposed a threshold-guided technique, called *STTrace*, which compresses trajectories based on the target compression rate and the *Synchronous Euclidean Distance* (SED) within each trajectory. The main drawback of these approaches is that they use a lossy approach to compress trajectories. Thus, information loss is inevitable, and applications built on top of the approaches are dependent on the accuracy of the compressed trajectory data.

3 Green GPS Tracking for MLSA

In this section, we discuss the proposed *event-based GPS tracking* approach, called EBT, for energy conservation in MLSAs. EBT comprises two components: *hybrid duty-cycle scheduling* and *trajectory reconstruction*. The first component utilizes a hybrid approach that combines the conventional static duty-cycle (SDC) approach and an event-based dynamic duty-cycle (DDC) approach, which uses a 3-axis accelerometer (G-sensor) to detect *turn events* and trigger GPS receivers to lock accordingly. Then,
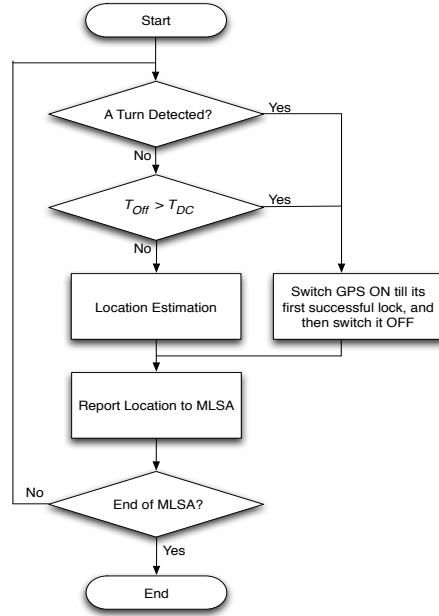
Fig. 1. The flowchart of the event-based GPS tracking (EBT) approach

the second component uses the sensory data of the G-sensor to estimate the location of the moving object and reconstruct the trajectory. The flowchart of the EBT approach is shown in Figure 1. Let T_{DC} be the length of the duty cycle in the embedded SDC, and let T_{Off} be the elapsed time since the last successful GPS lock event. We describe the two components in detail in the following subsections.

3.1 Hybrid Duty-Cycle Scheduling

The proposed hybrid duty-cycle scheduling approach is based on the intuition that, *to conserve energy, it is better to keep GPS receivers in the OFF mode, unless the location accuracy of the MLSAs is threatened*. The location accuracy of GPS tracking deteriorates in two situations: 1) the GPS receiver has been in the OFF mode longer than a pre-defined threshold period; and 2) there has been a significant change in the movement pattern of the object that the GPS receiver is tracking.

As shown in Figure 1, for the first situation, we define a threshold time T_{DC} [1], and employ an SDC-like approach that triggers a GPS receiver to lock whenever T_{Off} (i.e., the elapsed time since the last successful GPS lock event) reaches T_{DC}. To address the second situation, we introduce the *Turn Detection Algorithm* (TDA), which detects

---

[1] The value of T_{DC} depends on the location accuracy required by the MLSA.

'significant changes' (i.e., changes in direction) in the movement pattern and triggers GPS receivers to lock accordingly[2].

Specifically, TDA uses the *Poisson Sampling* approach to detect turn events at a rate of λ, i.e., the 'inter-sample' times follow an exponential distribution with a mean of $1/\lambda$. For each sample, TDA utilizes the G-sensor to read the acceleration on three axes. Then, using a sliding window of size w, it calculates the standard deviation of the acceleration in the direction that is orthogonal to gravity and the trajectory's orientation[3].

Let S be the set of the standard deviations stored in the history, let s_i be the i-th standard deviation in ascending order in S, and let α be the *target power saving ratio*. We set $S_{thresh} = s_j$, where $j = \lceil \alpha \times |S| \rceil$. Then, TDA identifies a turn event whenever the standard deviation in a sliding window is greater than the latest S_{thresh} value.

There are three ways to implement S in TDA:

1. *Infinite-Queue Approach* (IQA): S contains all standard deviations calculated in the previous sliding windows. The size of S is infinite.
2. *FIFO-Queue Approach* (FQA): The standard deviations calculated in the previous L sliding windows are stored in S in a First-In-First-Out (FIFO) manner.
3. *Duo-Queue Approach* (DQA): Similar to FQA, the size of S is limited to L, but DQA divides S into two subsets: one is used to store standard deviations that are smaller than or equal to S_{thresh}; and the other is used to store the standard deviations that are greater than the latest value of S_{thresh} when they are calculated. The sizes of the subsets are $\lceil \alpha L \rceil$ and $L - \lceil \alpha L \rceil$ respectively. We use the FIFO approaches to implement both subsets.

Note that to achieve the target power saving ratio α, it is advisable to increase the sampling frequency (i.e., a large λ value) and keep as many of the standard deviations calculated in the previous windows as possible (i.e., using the IQA scheme). However, the higher the sampling rate, the greater will be the computational resource required by TDA, and that may impact other applications on a smart phone. With regard to the suitability of the three approaches, IQA is not feasible because it is impossible to have an infinite queue in practice. DQA is more suitable than FQA, as it is better able to preserve the distribution of the standard deviations calculated in the previous windows. Consequently, it yields a better approximation of S_{thresh}.

3.2 Location Estimation

As shown in Figure 1, if no turn events are detected and the time since the last successful GPS lock event is less than T_{DC}, EBT enters the *Location Estimation* phase to approximate its current location without using GPS. In this study, we integrate EBT with the

---

[2] We do not use *digital compasses* to detect turn events because they are very sensitive to magnetic fields. We present a detailed evaluation in the following sections.

[3] For simplicity, in the experiment, we place the smart phone on the vehicle carefully, such that the Y-axis of the G-sensor is parallel to the vehicle's orientation, and the Z-axis of the G-sensor is parallel to gravity. Therefore, the calculation of the standard deviation in TDA only counts the accelerations on the X-axis of the G-sensor.
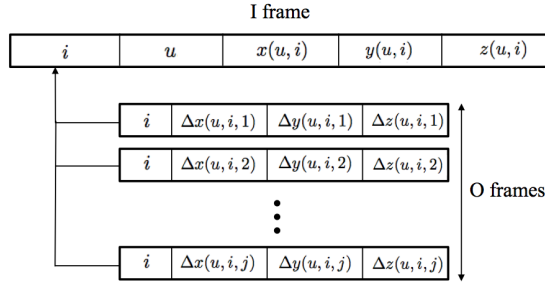
I frame

i	u	$x(u,i)$	$y(u,i)$	$z(u,i)$

i	$\Delta x(u,i,1)$	$\Delta y(u,i,1)$	$\Delta z(u,i,1)$
i	$\Delta x(u,i,2)$	$\Delta y(u,i,2)$	$\Delta z(u,i,2)$

$\vdots$

i	$\Delta x(u,i,j)$	$\Delta y(u,i,j)$	$\Delta z(u,i,j)$

O frames

Fig. 2. An example of I frames and O frames in the IFC scheme.

displacement measurement algorithm (DMA) [22], which uses accelerometers to measure the displacement of a moving object. Then, we implement the location estimation mechanism by combining the displacement measurement results and the location of the last successful GPS lock event.

DMA is comprised of six steps: interpolation, average filtering, de-shimmying, offset compensation, first integration, and second integration. We omit the interpolation step because we assume the granularity of the G-sensor's data is very fine. Moreover, to improve the integration of DMA with EBT and simplify the implementation, we set the window size used in the *average filtering* step equal to the sliding window size of TDA (i.e., w). The window size used in the *de-shimmying* step is $2w$.

4 Green Data Communications in MLSAs

Like the excessive power consumption of GPS tracking, the data communication overhead is a challenging issue in MLSAs. This is because MLSA data runs as a continuous stream, which is expensive in terms of bandwidth utilization, monetary expense, and power consumption. In this section, we propose a trajectory compression algorithm, called *Inter-Frame Coding* (IFC), to reduce the energy consumption of MLSAs in the data communication layer. The rationale behind IFC is that (1) *spatial and temporal localities are common on a trajectory*; and (2) *the spatial and temporal offsets between any two contiguous data points on a trajectory are limited to the object's mobility and the trajectory's sampling rate*. Thus, IFC exploits the spatial and temporal localities of contiguous spatio-temporal data to reduce the redundancy in the raw spatio-temporal data stream.

There are two types of data points in the IFC scheme: *I frames*, which contain the *index* data points of a trajectory; and *O frames*, which contain the *offsets* of the subsequent data points that correspond to the I frames, as shown in Figure 2. Let $\mathcal{I}_i^u$ denote the i-th I frame that represents the v-th spatio-temporal data point of the u-th trajectory (i.e., $\mathcal{T}_v^u$); and let $\mathcal{O}_{i,j}^u$ denote the j-th O frame associated with $\mathcal{I}_i^u$, i.e., the offset of $\mathcal{T}_{v+j}^u$ to $\mathcal{T}_v^u$.

Specifically, $\mathcal{I}_i^u = (i, u, x(u,i), y(u,i), z(u,i))$, where i is the sequence number of $\mathcal{I}_i^u$; u is the trajectory identifier; and $x(u,i)$, $y(u,i)$, and $z(u,i)$ are the longitude, latitude, and timestamp of $\mathcal{T}_v^u$ respectively. In addition, $\mathcal{O}_{i,j}^u = (i, \Delta x(u,i,j), \Delta y(u,i,j),$

$\Delta z(u, i, j)$), where i is the sequence number of the I frame associated with $\mathcal{O}_{i,j}^u$; and $\Delta x(u, i, j)$, $\Delta y(u, i, j)$, and $\Delta z(u, i, j)$ represent, respectively, the longitude, latitude, and time offsets of $\mathcal{T}_{v+j}^u$ to $\mathcal{T}_v^u$.

Generally, an I frame is associated with n O frames. The value of n, which is a tunable system parameter, depends on several factors, such as the sampling rate of the trajectory data, the speed of the moving object, and the data compression ratio required by the application (which we discuss in detail in the next subsection). However, when the offset values exceed the range allowed in an O frame[4], a new I frame must be created, even though the number of O frames associated with the old I frame is less than n.

We assume that 1) the maximum possible speed of a moving object along the lines of latitude and longitude is V_{max} meters per second; 2) the trajectory data is collected at a rate of s data points per second; 3) the maximum value of the latitude and longitude offsets is MAX_{dist}; and 4) the maximum value of the time offset is MAX_{time}. Since we know that $V_{max} \times \frac{n}{s} \leq MAX_{dist}$ and $\frac{n}{s} \leq MAX_{time}$, we can obtain the upper bound of n as follows:

$$n \leq Min \left(\frac{MAX_{dist} \times s}{V_{max}}, MAX_{time} \times s \right). \tag{1}$$

In addition, we define the *compression ratio* Ψ as the ratio of the data using the IFC scheme over the data without using the scheme, i.e., the raw data. Since the size of an I frame is $Size\_I$, the size of an O frame is $Size\_O$, and each I frame is associated with n O frames at most, we can obtain the value of Ψ by

$$\Psi = \frac{Size\_I + n \times Size\_O}{Size\_I \times (n+1)} = \frac{Size\_O}{Size\_I} + \frac{Size\_I - Size\_O}{Size\_I \times (n+1)}. \tag{2}$$

Thus, the higher the value of n, the smaller will be the compression ratio Ψ. The upper bound of the compression ratio is $\frac{Size\_O}{Size\_I}$ when n approaches infinity. However, we find that a very large n value is infeasible in the IFC scheme. There are two reasons for this phenomenon. First, when n is very large, the O frames are used to store most spatio-temporal data points. This is computationally expensive for spatio-temporal applications because (i) each data query in the application layer involves two separate database queries; and (ii), restoring the O frames to the raw data format requires a tremendous amount of computation. Moreover, if n is very large, the loss of a single I frame may result in the loss of the original data points because the I frames are crucial for restoring the original data points of the O frames.

The second reason is that, the larger the value of n, the greater the likelihood that an I frame will have less than n O frames in the database. In other words, when n is large, the subsequent n points of an I frame are more likely to have oversized offset values that can not be represented by O frames. As a result, a new I frame must be created, even though the number of O frames associated with the former I frame is less than n. Hence, the IFC scheme cannot achieve the theoretical compression ratio when n is large.

---

[4] For instance, the offset values may exceed the range allowed in an O frame if, in the data stream, a long period containing data points is missed due to transmission errors or GPS errors.

5 Evaluation

In this section, we evaluate the proposed solution for MLSA GPS tracking on a real dataset. The data was collected by volunteers during 50 trips on the TPE-CMS bus system [20] using the VProbe application [23] on their smart phones (including Acer Liquid, HTC Magic, Samsung Nexus S, and Sony Ericsson XPERIA X10 phones). Specifically, the sensory data includes GPS trajectories, digital compass directions, and 3-axis accelerations. The sampling rate for the data is 20Hz. We use the dataset as the ground truth and perform trace-based simulations. Unless otherwise specified, the maximum time allowed between two GPS locks (i.e., T_{DC}) is 60 seconds. The Poisson sampling rate used in TDA (i.e., λ) is 1 Hz; the target power saving ratio (i.e., α) is 0.95; the queue length in the three queue management schemes (i.e., L) is 1000; and the size of the sliding window (w) is 50 samples. All the results are based on the average performance of 10 simulation runs. We discuss the evaluation results in the following subsections.

5.1 Evaluation of Turn Event Detection

First, we use the realistic dataset to evaluate the feasibility of using digital compasses to detect turn events in MLSAs. The dataset contains trajectories of 86,607 seconds in total, and there are 235 turn events that are marked manually as ground truth. The digital compass-based approach reported 795 turn events. Among them, 115 turn events are detected correctly (i.e., the accuracy of turn event detection is 115/235=48.94%), and 680 events are false-alarms (i.e., the false positive ratio is 680/795=85.53%). The reason is that digital compasses are very sensitive to environmental factors, especially when they are placed in a vehicle with diverse magnetic fields. Thus, digital compasses may not be feasible for detecting turn events in general MLSA scenarios.

Next, we evaluate the proposed TDA scheme in terms of the hit rate of turn detection, with different values of α under EBT with the IQA, FQA, and DQA queue management schemes. The results are shown in Figure 3. We observe that EBT's hit rate is more than 98% in all test cases when the TDA scheme is used (i.e., $\alpha < 1$). In contrast, the hit rate is approximately 71% when $\alpha = 1$ (i.e., under the SDC scheme). There are no significant differences between the hit rates of the IQA, FQA and DQA schemes.

5.2 Evaluation of Event-based Tracking

We evaluate EBT's performance in terms of its power saving ratio and location accuracy in MLSA GPS tracking. Using the dataset of 50 bus trips, we compare the power saving ratio (α') achieved with different target power saving ratios (α) under EBT with the three queue management schemes. From the results, shown in Figure 4, we observe that α' increases with α, and the IQA and DQA schemes perform better than FQA. The reason is that FQA implements the FIFO queue with a size limit of $L = 1,000$, and the selected threshold S_{thresh} is not usually representative of the true distribution of turn events; hence, there is a large number of false alarms (i.e., false positive turn events). Meanwhile, although the values of α achieved by DQA are slightly lower than those
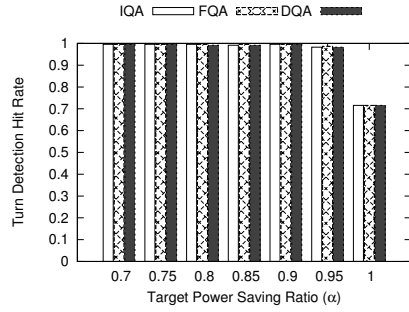
Fig. 3. Comparison of turn detection hit rates with different target power saving ratios (α) under EBT with the three queue management schemes
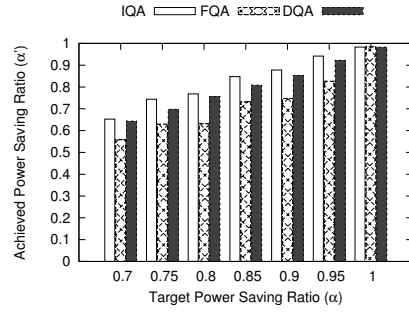

Fig. 4. Comparison of the power saving ratio (α') achieved with different target power saving ratios (α) under EBT with the three queue management schemes
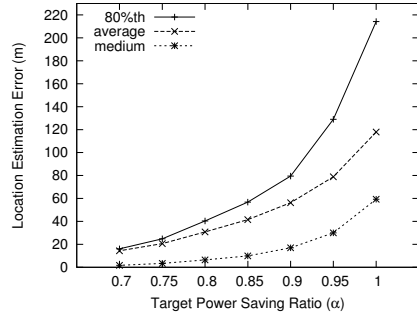

Fig. 5. Comparison of the location estimation errors using different target power saving ratios (α) under EBT with the DQA scheme
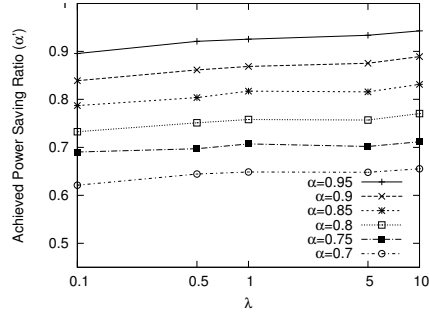

Fig. 6. Comparison of the power saving ratio (α') achieved with different sampling rates (λ) and target power saving ratios (α) under EBT with the DQA scheme

under IQA in all test cases, we consider that the DQA scheme is a feasible solution because its power saving performance is comparable to that of the IQA scheme, and it is a practical solution that does not depend on an infinite queue.

In Figure 5, we compare the location estimation errors with different α values under EBT using the DQA scheme. The distance error increases with α because the loss of location accuracy is the price paid (i.e., the trade-off) for reduced energy consumption. The results demonstrate that the implementation of the TDA scheme improves the location accuracy significantly in EBT. The average location estimation error is about 120m when the TDA scheme is not applied (i.e., $\alpha = 0$). By contrast, it is about 80m (i.e., a 33% improvement) with the TDA scheme and $\alpha = 0.95$. We observe that the distribution of location estimation errors is skewed towards instances with large values. Even so, TDA's overall performance is considered acceptable because the median error rate is less than 30m when $\alpha = 0.95$, and less than 10m when $\alpha < 0.85$.
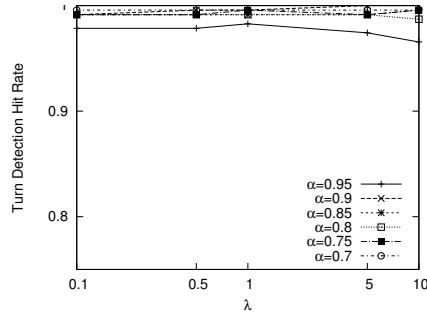
Fig. 7. Comparison of the turn detection hit rates with different sampling rates (λ) and target power saving ratios (α) under EBT and the DQA scheme
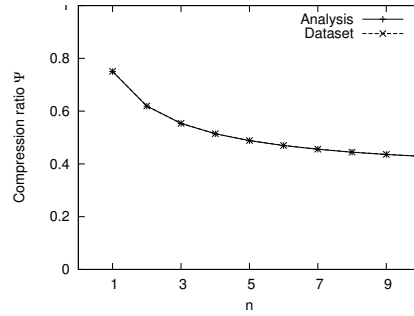
Fig. 8. Comparison of the theoretical compression ratio and the compression ratio achieved using the realistic dataset under the IFC scheme with different n values

We also evaluate the impact of MLSA sampling rates (λ) on the power saving performance and turn detection performance of GPS tracking in MLSAs. The results, shown in Figure 6, demonstrate that EBT achieves a good power saving ratio that is slightly lower than the target ratio when the value of λ is between 0.1 Hz and 10 Hz, which is the range of typical sampling rates in MLSAs. Moreover, the results in Figure 7 show that EBT can detect nearly all the turn events under different λ values.

5.3 Evaluation of Inter Frame Coding

Finally, we evaluate the proposed MLSA data communication solution, i.e., the IFC scheme, in terms of the data compression ratio. We implement the IFC scheme using the open-source PostgreSQL database (version 8.4.4) [24] and the PostGIS spatial database extension (version 1.5.1) [25]. For I frame data in the IFC scheme, we use the *Point* data type (16 bytes) to store the location information (x and y), the *Timestamp* data type (8 bytes) to store time information (z), and the *Integer* data type (4 bytes) to store sequence numbers (i) and trajectory identifiers (u). For the O frames, we use the *Integer* data type to store the I frame sequence number (i); and the *Short Integer* data type (2 bytes) to store the longitude, latitude, and time offsets Δx, Δy, and Δz. Therefore, the size of an I frame ($Size\_I$) is 32 bytes, and that of an O frame ($Size\_O$) is 10 bytes. Note that the distance offsets (Δx and Δy) are measured in meters[5], and the time offset (Δz) is measured in seconds.

In Figure 8, we compare the theoretical compression ratio (cf. Equation 2) and the ratio achieved in the realistic trajectory dataset with different n values under the IFC scheme. We observe that the two curves are nearly overlapped completely, and the compression ratio is lower than 0.5 after the value of n becomes larger than 4. Moreover, using the same dataset, we compare IFC with five trajectory compression schemes: OPW [18], TDTR [18], STTrace [19], Uniform Sample [19], and Douglas-Peucker (DP) [21].

---

[5] For simplicity, we approximate the distance between two data points along lines of latitude and longitude by one meter every 10^{-5} degrees, which is valid at sea level on the Equator.
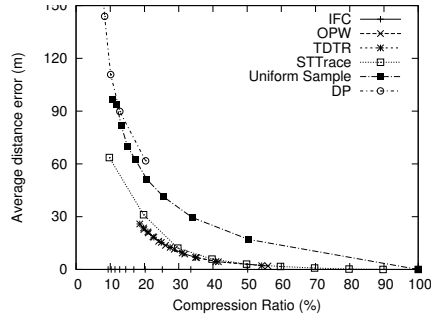
Fig. 9. Comparison of the results of the IFC, OPW, TDTR, STTrace, Uniform Sample, and DP schemes in a two-objective space (i.e., the average distance error and the compression ratio) under various configurations

For each scheme, we use an exhaustive set of configurations[6], and observe their *Pareto frontiers* [26] between the average distance error and the compression ratio, as shown in Figure 9.

Figure 9 shows that the IFC scheme outperforms the other schemes significantly and it always achieves the "Pareto optimum" [26]. In other words, there are no instances on the other schemes' frontiers that have smaller distance errors and lower compression ratios than the instances on IFC's frontier. The results also demonstrate that the IFC scheme is lossless during MLSA data compression, as its distance error is zero despite the different compression ratios achieved. In contrast, the other schemes are lossy during data compression. The lower the compression ratio, the larger will be the distance error.

6   Conclusion

We propose a two-layer approach to reduce the energy consumption of emerging mobile location sensing applications (MLSAs). Specifically, on the MLSA GPS tracking layer, our event-based GPS tracking approach (EBT) combines the strengths of static and dynamic duty-cycle scheduling approaches; and on the MLSA data communication layer, our Inter-Frame Coding (IFC) approach exploits the spatio-temporal localities in MLSA data compression. We evaluate the proposed schemes via a comprehensive set of simulations based on a real dataset. The results demonstrate that the EBT scheme can achieve a good power saving ratio while maintaining the location estimation error within an acceptable range. In addition, the results show that, compared to existing schemes, the IFC scheme is lossless and effective in MLSA data compression. The solution proposed in this study is simple and effective, and it is generalizable to other mobile location sensing applications.

---

[6] For instance, we vary the number of O frames per I frame (i.e., n) from 1 to 10 in the IFC scheme; the sampling gap from 1 to 10 in the Uniform Sample scheme; the target compression ratio from 10% to 90% in the STTrace scheme; and the error threshold from 10m to 90m in the OPW, TDTR, and DP schemes.

References

1. Kirkpatrick, M.: Why We Check In: The Reasons People Use Location-Based Social Networks. ReadWriteWeb.com (June 28 2010) Retrived June 1, 2011.
2. Miluzzo, E., Lane, N.D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S.B., Zheng, X., Campbell, A.T.: Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe applications. In: ACM SenSys. (2008)
3. Eisenman, S.B., Miluzzo, E., Lane, N.D., Peterson, R.A., Ahn, G.S., Campbell, A.T.: BikeNet: A mobile sensing system for cyclist experience mapping. ACM Transactions on Sensor Networks **6**(1) (2009) 6:1–6:39
4. Huang, J.H., Amjad, S., Mishra, S.: Cenwits: A sensor-based loosely coupled search and rescue system using witnesses. In: ACM SenSys. (2005)
5. Zhang, P., Sadler, C.M., Lyon, S.A., Martonosi, M.: Hardware design experiences in ZebraNet. In: ACM SenSys. (2004)
6. Sistla, A.P., Wolfson, O., Chamberlain, S., Dao, S.: Modeling and Querying Moving Objects. In: IEEE ICDE. (1997)
7. Constandache, I., Choudhury, R.R., Rhee, I.: Towards Mobile Phone Localization without War-Driving. In: IEEE Infocom. (2010)
8. Jurdak, R., Corke, P., Dharman, D., Salagnac, G.: Adaptive GPS duty cycling and radio ranging for energy-efficient localization. In: ACM SenSys. (2010)
9. Kim, D.H., Kim, Y., Estrin, D., Srivastava, M.B.: SensLoc: sensing everyday places and paths using less energy. In: ACM SenSys. (2010)
10. Kjargaard, M.B., Bhattacharya, S., Blunck, H., Nurmi, P.: Energy-efficient Trajectory Tracking for Mobile Devices. In: ACM MobiSys. (2011)
11. Kjargaard, M.B., Langdal, J., Godsk, T., Toftkjar, T.: Entracked: energy-efficient robust position tracking for mobile devices. In: ACM SenSys. (2009)
12. Paek, J., Kim, J., Govindan, R.: Energy-efficient rate-adaptive GPS-based positioning for smartphones. In: ACM MobiSys. (2010)
13. Sundaresan, S., Koren, I., Koren, Z., Krishna, C.M.: Event-driven adaptive duty-cycling in sensor networks. International Journal of Sensor Networks **6**(2) (October 2009) 89–100
14. Youssef, M., Yosef, M.A., El-Derini, M.: GAC: Energy-Efficient Hybrid GPS-Accelerometer-Compass GSM Localization. In: IEEE Globecom. (2010)
15. Zhuang, Z., Kim, K.H., Singh, J.P.: Improving energy efficiency of location sensing on smartphones. In: ACM SenSys. (2010)
16. Salomon, D.: Data Compression: The Complete Reference. 4th edn. Springer (December 2006)
17. Gudmundsson, J., Katajainen, J., Merrick, D., Ong, C., Wolle, T.: Compressing spatio-temporal trajectories. In: ICAC. (2007)
18. Meratnia, N., de By, R.A.: Spatiotemporal Compression Techniques for Moving Point Objects. In: EDBT. (2004)
19. Potamias, M., Patroumpas, K., Sellis, T.: Sampling Trajectory Streams with Spatiotemporal Criteria. In: SSDBM. (2006)
20. Lin, C.Y., Chen, L.J., Chen, Y.Y., Lee, W.C.: A comfort measuring system for public transportation systems using participatory phone sensing. In: PhoneSense. (2010)
21. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer **10**(2) (1973) 112–122
22. Chen, T., Hu, W., Sun, R.: Displacement Measurement Algorithm Using Handheld Device with Accelerometer. In: Asia-Pacific Conference on Wearable Computing Systems. (2010)
23. : VProbe: sensing and probing your driving experience. http://vprobe.org/
24. : PostgreSQL Database. http://www.postgresql.org/
25. : PostGIS. http://postgis.refractions.net/
26. Fudenberg, D., Tirole, J.: Game Theory. The MIT Press (Auguest 1991)