EMPAC:Exploiting Movement-Pattern for Collaborative Localization in Mobile Networks

Zhenzhen Tian
StuID:5090309019

***Abstract*—Statement, this paper contains not only the work I do during the prp project, but the work inspired during Professor Wang's Wireless Communications Principles Applications, After the prp project(given the score of the prp), I add the Effect of Shadow Fading into the project, and the Entropy part is also the work I do after the prp's score is given. In the field of sensor networks, localization is very necessary for the effective use of ad hoc sensor networks. In the localization field exists many methods. For many reasons, both technical and AE(application environmental), such methods are often perform badly without taking a consideration of the movement-pattern of the mobile nodes.**

In this paper, we present the EMPAC(Exploiting Trace-Pattern Collaborative Localization for Mobile Networks) algorithm for the application environment that contains mobile nodes with different movement-patterns. In EMPAC, each node estimate his location not only from his own track sensor devices(3D accelerometer, electronic compass, etc.) and neighbors it encounters but also combine with his own movement-pattern. We evaluate the EMPAC performance both through simulation and the real world traces from 08,09 infocom's five traces[1][2].

I. INTRODUCTION

As the application of mobile sensor network plays a significance role in a wide variety of domains such as the environment monitoring, vehicle tracking etc. Location awareness plays a significant role in all those applications. In most of the application situation the mobile nodes are often with different movement-patterns, for instance, a students in university will be more willing to choose the way to classroom that he familiar with while some one may just moving like idiots choosing their way randomly. In this case, if they are treated without exploiting their movement-patterns, the performance will be worse than the case considered. Besides, in some area such as the urban canyons or underground, the GPS(global positioning system) signal could not reach the node, the node's neighbor(or multi-hop neighbors) with a good position estimation, however, may be able to help the node to estimate his location. Furthermore, the GPS is cost prohibitive.

To solve these problems, several mobile networks make use of a small number of seed nodes configured GPS to move around the area and refresh the node's location. if the node didn't come arose the seeds the localization is impossible, unfortunately, such case is unavoidable. If we use fixed location beacons as localization references. The beacons should be able to cover all the area of the networks which will be cost prohibitive.

In this paper, we present the EMPAC(Exploiting Movement-Pattern for Collaborative Localization in Mobile Networks) algorithm. it is a distributed localization algorithm to enable each node to estimate their location more quickly and accurately, what's more, the algorithm could be implemented by the node configured with some small SCM(single chip micyoco). Each node not only make use of his movement-pattern introduced in Section II, but also exchange his location estimation with his neighbors. During the period of the disconnection the node can also predict the location estimation using a DR(dead-efficient) system[10][11].

In the subsequence sections, we evaluate our EMPAC method against LOCALE[3] that use collaborative localization estimation methods to evaluate location for mobile networks. our algorithm, EMPAC, compared with the LOCALE, not only exploits the neighbors location information for localization, but also makes use of the node's movement-pattern and the DR(dead-reckoning) system to make a more accurate localization. the significance of EMPAC is that it regards different nodes with different attitudes based on their movement-patterns, the nodes with obvious movement-patterns could help those without refine their location estimations. This hypothesis is more suitable for the real world case, for instance, in a city, there are not only some cars moving on the road without obvious movement-patterns, such as taxi that the drivers have to drive the taxi based on the desire of his customer, but also some ones with a significance movement-pattern, such as the buses whose route is scheduled before.

This paper is organized like this: we introduce our localization algorithm in details in Section II. Section III is the simulation. Section IV is the conclusion.

II. EMPAC

The EMPAC(Exploiting Movement-Pattern for Collaborative Localization in Mobile Networks) is a distributed algorithm in which the node not only estimates his own location based on the neighbor encountered , but also refine his location estimation based on his movement-pattern no matter whether a neighbor existing not not.

This subsection is organized like this: subsection A is the Path loss modeling Effect of Shadow Fading, subsection B introduces how to exploit movement-pattern for localization in mobile networks. Subsection C introduces the collaborative localization for mobile networks. Subsection C introduces how to merge the two strategies' location estimation into a more accurate one.

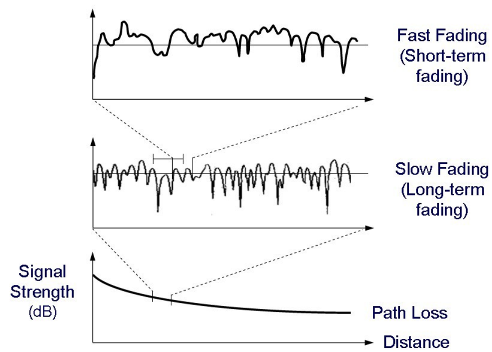

Fig. 1. The Fading of the signal propagation

A. Path loss modeling Effect of Shadow Fading

When the mobile nodes's signal propagating in the outdoor environment, its will suffer a loss of signal strength related to the distance between two mobile node that share the location information shown in Fig.1 For the reason that even the node form the same distance, their received signal strength will be different, the process could be The process of signal propagation could be modeled be the Shadow Fading(slow fading).

$$L_p = L_0 + 10\alpha log(d) + X \quad (1)$$

X is a random variable with a distribution that depends on the fading component. Based on measurements and simulations, this variation can be expressed as a log-normal distributed random variable. The problem of shadow fading is that all locations at a given distance may not receive sufficient signal strength for correctly detecting the information.

B. Exploiting Movement-Pattern

Assume that at t_0 a node, N_1, is located in

$$X_0 = \binom{x_0}{y_0} \quad (2)$$

Since any node's velocity is finite.

$$v < v_{max} \quad (3)$$

v_{max} is the maximum velocity all the noded can reach.

After a time slot Δt, the N_1's position is limited to a certain range by:

$$\sqrt{(X_1 - X_0)^T \times (X_1 - X_0)} < v_{max}\Delta t \quad (4)$$

The X_1 is the position of N_1 at $t_0 + \Delta t$

In most application, the node's moving range is limited, in city, for example, a taxi will not go out of the city, because no one will choose taxi as the means for a business trip and the people will often moving within the city. In our research, our Localization method is designed for those mobile nodes whose movement range is confined within a certain range, for instance, the car or human beings etc. living in New York city or just in a university campuses (NCSU and KAIST). We divide the Node's movement range into several grids, each grid is a rectangle: $(v_{max}\Delta t) \times (v_{max}\Delta t)$, for instance, the KAIST is divided into 50×70 grids, shown in Fig. 2.

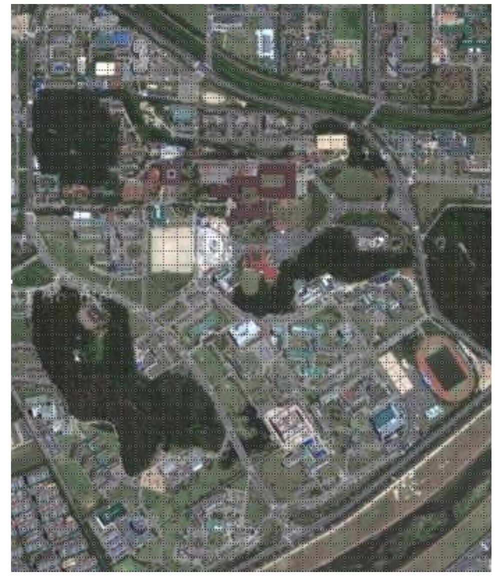

Fig. 2. The KAIST campus is divided into 50×70 grids

We use the Map_Matrix M_{ij} to represent the field we concern, the element $M(3, 4)$ corresponding to the grid in row $i = 3$ and column $j = 4$. Each node's trace could be described by a set of Map_Matrix$\{M_{t0}, M_{t1}, M_{t2}, etc.\}$. In real world case, people typically keep a routing of visiting the same places every day such as going to an office, but at the same time, make irregular trips. It is not the case where people would always randomly choose places to visit and visit them in a random order. there exists many works[12][13][14] about the regularity of daily trip patterns of humans. None of the existing work reflects the movement-patterns appearing in real human walk traces. Since a node's moving process is a markov chain, so we use a state transmission matrix to represent the process that a node moving from a gird into another. For illustration purpose we use a Map_Matrix $M(2 \times 2)$ to represent a field shown in Fig. 3.

- Initially: $\pi_1 = [1 \ \ 0 \ \ 0 \ \ 0]$
- One Step: $\pi_2 = \pi_1 \times P = [0.3 \ \ 0.2 \ \ 0.5 \ \ 0]$
- k Step: $\pi_k = \pi_1 \times P^k$

π is corresponding to M, the only difference is that π is gotten by transmit the M from a matrix size 2×2 into the matrix size 1×4. The key point of this process is P(state transmission matrix). For any node we can got the P by statistical methods, for instance, a node's trace gotten by GPS, could be divided by the grids into several segments, then we count the times of the grid jump from grid $M(i, j)$ to grid $M(i + 1, j)$(jump to the adjacent east grid), $M(i, j + 1)$(jump to the adjacent north grid), $M(i-1, j)$ (jump to the adjacent west gird), M(i, j-1)(jump to the adjacent south grid), $M(i, j)$(staying in the same grid).

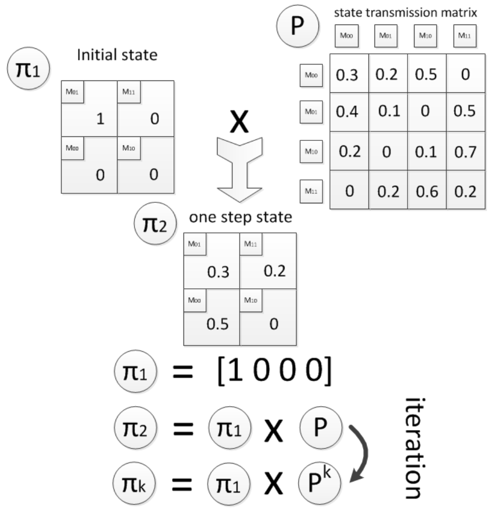

Fig. 3. The process of state transmission

$$N_e = JT\langle M(i,j) \rightarrow M(i+1,j)\rangle \quad (5)$$
$$N_s = JT\langle M(i,j) \rightarrow M(i,j-1)\rangle \quad (6)$$
$$N_n = JT\langle M(i,j) \rightarrow M(i,j+1)\rangle \quad (7)$$
$$N_w = JT\langle M(i,j) \rightarrow M(i-1,j)\rangle \quad (8)$$
$$N_m = JT\langle M(i,j) \rightarrow M(i,j)\rangle \quad (9)$$

$$N_t = N_e + N_n + N_w + N_s + N_m \quad (10)$$

$$P_e = \frac{N_e}{N_t} \quad (11)$$
$$P_s = \frac{N_s}{N_t} \quad (12)$$
$$P_n = \frac{N_n}{N_t} \quad (13)$$
$$P_w = \frac{N_w}{N_t} \quad (14)$$
$$P_m = \frac{N_m}{N_t} \quad (15)$$

$JT\langle M \longrightarrow \acute{M}\rangle$ is defined to calculate the node's Jump Times from M to $\acute{M}$, so the N_n means the times of a node jump from current grid $M(i,j)$ to the north adjacent gird $M(i,j+1)$. For example if the times of a node jump from $M(3,3)$ to $M(3,4)$ is 3, them $N_n = 3$. After the statistical methods, we can get the state transmission matrix P.

For real case, we can record the trace by GPS for several days, the record data could act as a training trace. With the help of statistical methods mentioned above, we can get the P, then we can predict the behavior of node(localization) of the coming days.

Obviously, this algorithm will perform well in the case that the node's movement-pattern is very obvious. for some node, however, their movement-pattern may not be very obvious. To solve this problem, we introduce the collaborative localization method to improve the Localization accuracy of those whose movement-pattern is not so obvious with the help of the node with the obvious movement-pattern.

C. Collaborative Localization

This section is mainly talking about how to collaborative exchange one node's location estimation with the neighbors encountered to help improve those node without a accurate localization estimation. This part are inspired by[3]. This section can be divide into 2 subsections. Subsection 1) introduce represent a location with the location estimation(mean) and the certainty(variance). Subsection 2) introduce how to exchange a node's location information with it's neighbors.

1) Location Representation: In real world case, if we equipped the node with a three dimensional accelerometer[10] and a electronic compass[11], we can got a approximation location of the mobile node. Since those equipments are very inaccurate, the uncertainty about the location will grow very fast with respect to time.

In probability theory, the CLT(central limit theorem) states, that given certain conditions, the mean of a sufficiently large number of independent random variables, each with finite mean and variance, will be approximately normally distributed. Based on CLT we use location estimation(mean) and a certainty(covariance) to represent a node current location information.

In the 2-dimensional case the probability density function of location estimation vector is:

$$P(X) = \frac{1}{2\pi\sqrt{|C|}} e^{-\frac{1}{2}(X-\mu)C^{-1}(X-\mu)} \quad (16)$$

$$\mu = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix} \quad (17)$$

$$C = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix} \quad (18)$$

the μ(16) represent location estimation and The parameter C(covariance)(17) represent location certainty, we can see that to define the equation only two parameters (16)(17), C for certainty and μ for location estimation, is necessary.

2) Exchange Location Information with Neighbor Node: As mentioned before, our algorithm is distributed. So the individual's coordinate is different from each other.

To solve this problem, the coordinate transmission process is necessary before they merge neighbor's location estimation, the N_h represents Host node, and N_n represents Neighbor node. Their coordinate different is shown in Fig. 3.

To define the distribution and to be able to merge the location information from a neighbor node, seven parameters are necessary: $1st$ the location estimation μ, $2nd$ the covariance matrix C, $3rd$ the distance d between the host and the neighbor nodes[15][16][6][3], $4th$ the angle α of arrival signal orientation from the neighbor node[5], $5th$ the coordinate systems intersection angle β between host coordinate and neighbor coordinate, $6th$ the distance D between two coordinates, $7th$

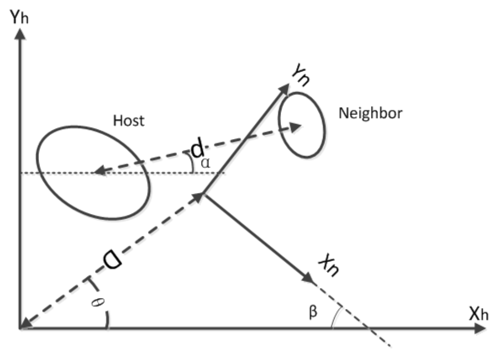

Fig. 4. The representation of Host and Neighbor nodes with different coordinates

the angle θ between D and *Host's*. the last three parameters β ,D and $\theta$$ could be gotten by set two landmarks, with two landmarks locations in each coordinate the β, D and θ could be deduce very easily.

With the help the localization estimation from the neighbor, a observed μ_{ob} could be gotten by:

$$\mu_{ob} = \left(\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \times D + R(\beta) \times \mu_n - \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \times d \right) \quad (19)$$

With the help of the neighbor, a observed C_{ob} could be gotten by:

$$C_{ob} = R(\beta)^T \times C_n \times R(\beta) \quad (20)$$

The rotation matrix is defined as:

$$R(\beta) = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \quad (21)$$

The node localization accuracy could be improved by merge the host node location information and the transformed location information from the neighbor node. Due to the subjective(localization devices) and objectives(environment) reasons each node has estimations with different certainties(covariance), Therefore, we combine the estimation with respect to their certainties(covariance) acting as the weight. Our merging methodologies are inspired by prior robotics work by Smith and Cheeseman[4].

The merged certainty is calculated by

$$C_{merged} = C_h - K \times C_h \quad (22)$$

The merged estimation is calculated by

$$\mu_{merged} = \mu_h + K \times (\mu_{observed} - \mu_h) \quad (23)$$

The K, used above, is the merge factor defined as

$$K = C_h \times (C_h + C_{observed})^{-1} \quad (24)$$

So far, we have gotten two sets of location estimation information(Exploiting Movement-Pattern and Collaborative Localization)

D. Merge Strategy

Now, we have two different kinds of location estimation information, one is the *Exploiting Movement-Pattern Localization Estimation* described in section II.A, another is the *Collaborative Localization* described in section II.B. Those two kinds of location information are very different from each other, *Exploiting Movement-Pattern Localization Estimation* is discrete, while *Collaborative Localization* is continuous. There are two strategies to merge them into one location estimation information.

1) transform the discrete location information into a continuous one by interpolation method.
2) transform the continuous location information into a discrete one by sampling method.

Since, our algorithm is distributed, which should be able to be implemented on the SCM(Single Chip Micyoco). The first strategy is not suitable for SCM's limited memory. So we choose the second method(sampling).

In section II.A, we use function (15) to describe the location information, we sample this location estimation distribution by:

$$M(i,j) = P(X) \quad where \quad X = \begin{pmatrix} i \cdot v_{max}\Delta t \\ j \cdot v_{max}\Delta t \end{pmatrix} \quad (25)$$

After the process of sampling, the value of $M(i,j)$ is still continuous, which is still hard for the implementation on the SCM. So we introduce the UQP(Uniform Quantization Process). In the UPQ we quantized the probability density into $N = 2^\nu$ levels, the ν is the bit number to store the quantized value. Then the length of each quantization region is:

$$\Delta = \frac{1}{N} = \frac{1}{2^\nu} \quad (26)$$

The quantized values are the midpoints of the quantization regions.

Since the distribution is designed for the discrete location estimation distribution, the UPQ is also suitable for*Exploiting Movement-Pattern Localization Estimation* described in section II.

After this process, we got two quantized discrete location estimation distributions. In order to merge the two kinds of location estimation information. we use the **median percent area error**, used in LOCALE localization method presented by[3]. As shown in Fig. 4, the median area error is the area of the smallest circle that includes 50% certainty of the probability.

After the definition of **median percent area error**, we can see that, when a node's certainty of his location estimation is higher, his **median percent area error** should be smaller. From a rule of thumb, we use the reciprocal as its weight of certainty, the **Certainty Weight** w is defined as:

$$w = \frac{1}{C} \quad (27)$$

C: defined as the grid number within the **median percent area error** in red circle shown in Fig. 4. With two distributions, the M_{mp}, the subscript mp is movement_pattern for

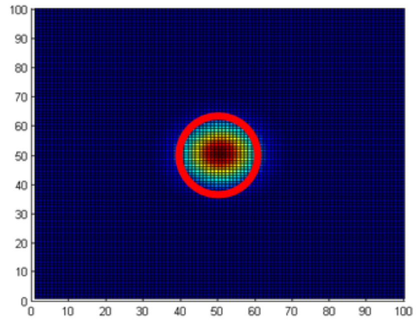

Fig. 5. The two distribution marked with the **median percent area error** in red circle

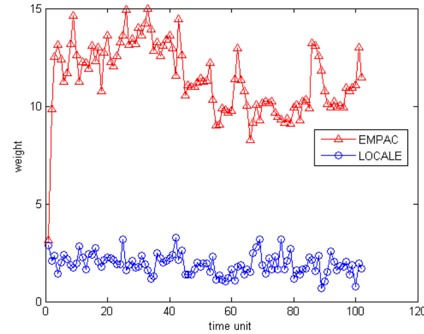

Fig. 6. Simulation arithmetic comparison between EMPAC(blue) and LOCALE(red) after 500 time slots

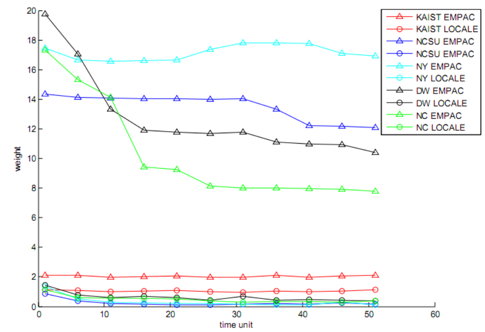

Fig. 7. Simulation arithmetic comparison between EMPAC(blue) and LOCALE(red) after 500 time slots

short and the M_{cl}, cl is collaborative_localizaiton for short, two weights, w_{mp} and w_{cl} we can merge them by calculate their weight average:

$$M_{merge} = \frac{w_{mp} \cdot M_{mp} + w_{cl} \cdot M_{cl}}{w_{mp} + w_{cl}} \tag{28}$$

After the merge process, the host's M_{mp} is refine by M_{merge}. This process is the key process to increase the accuracy of the localization.

III. Simulation

To evaluate the performance of the EMPAC, we perform our algorithm both with the simulated data and the data of human mobility traces from five different sites-two university campuses(NCSU and KAIST), New York City, Disney World(Orlando), and North Carolina state fair[1][2].

In this section, firstly we introduce our simulated data and analysis it with respect to time unit and *weight* defined in (26). Then we use the real world data and analysis it with respect to time and *weight*. Lastly, we simulate our algorithm with respect to the *normalized entropy* $\bar{H}$ defined in(28) and *weight* w.

In these simulations, we compare the EMPAC with the LOCALE introduced by [3] whose performance is all most $64\times$ better than just using beacon-and-DR method. In order to make a fair comparison, the infrastructure is the same, we place one fixed beacon in the center of the field, in [3] they place a fixed GPS beacon to help the encountered nodes to refine their location information.

A. Algorithm Performance by Simulated Data

This section's simulation is perform in the virtual world, we simulation a field with 50×50 grid, each time slot, the grid could only move to its adjacent grid. As described in (3), if $10min$ is selected as the time slot Δt and $6m/s$ is selected as the maximum velocity, the size of each grid is $(6 \times 10 \times 60) \times (6 \times 10 \times 60) = 3.6 \times 3.6km^2$ thus the field size of our simulation is $180 \times 180km^2$. In such a field, we placed 50 mobile nodes. To begin with, we generate the state transmission matrix for each node, in other word, we endow each node with a movement-pattern. Then we generate the nodes' traces based on their movement-patterns for 500 time slots. After this, we have the a movement-pattern and the the node's real trace data. To simulate the node's track sensor devices inaccurate properties, we introduce a random uncertainty to each node, it will increase the covariance(C) based on the the degree of the uncertainty, the localization estimation is also drifted based on the ΔX of the real trace and the uncertainty, the bigger the uncertainty is the greater drift the node localization estimation would be. In order to make a quantitative comparison. We use our metric: w(weight) defined in (26). For the 500 time slots, we extract the nodes w every 5 time slots and then calculate the arithmetic average $\bar{w}$. This results are shown in Fig. 5. From the Fig. 5, we can see that the EMPAC is almost $5\times$ better than the LOCALE.

B. Algorithm Performance by Real World Data

For the real world mobile nodes, we used the data in[1][2]. The data is recorded like this: The GPS receivers take reading of the volunteers' current positions at every 30 seconds. Each file represents a daily trace from one participants. One participant can make one or more daily trace files. But we cannot tell which files come from the same person The five traces are not the ideal traces we want, because the CLT assumption that the trace should be independent is not fully hold. But we can see that even though under this condition, the EMPAC is still perform better than the LOCALE. Here we regard the trace data collected at different periods as different

TABLE I
THE NORMALIZED ENTROPIES OF THE FIVE TRACES

traces	KAIST	NCSU	NY	DW	NC
$\bar{H}$	0.51	0.23	0.14	0.18	0.24

traces collected in the same period. for instance, students A's trace was collected on $Sept.27^{th}$ and $Sept.29^{th}$ we regard the two traces files as two different mobile node's recorded in the same day such as $Sept.28^{th}$, so the two traces regarded as different traces must be highly correlative.

Both of the EMPAC and the LOCALE rely on mobile node movement to propagate beacon information to other nodes with the assumption of CLT (central limit theorem) which require the nodes to be independent, each with finite mean and variance. The trace is collected by 4 students living in the campus dormitory at different period. As the data set can not tell which files come from the same person, the assumption of CLT does not fully hold. We use the data to see whether our algorithm could improve the performance in such a network. To begin with, we use the statistical method discussed by functions $(4 \sim 14)$ to generate the state transmission matrix. With the matrix and the real trace, we can realize our algorithm. The result is shown in Fig. 7. From the result, we can see thatthe EMPAC is approximate 3. $5\times$ better than the LOCALE.

C. Effect of movement-pattern

We now explore the performance of the EMPAC under varying node's movement-patterns. To begin with, we define the *normalized entropy* to represent the degree of our node's movement-pattern, defined as:

$$H(i,j) = \sum_{x=e}^{m} P_x log_5 \frac{1}{P_x} \tag{29}$$

The subscript x corresponding to the set $x = \{e, s, n, w, m\}$ $(10 \sim 14)$. We use base 5 to normalized the H(i,j). With the definition of H(i,j) we got the node's *normalized entropy* by calculated the average value $\bar{H}$ within the field.

$$\bar{H} = \frac{\sum_{j=i}^{i_{max}} \sum_{j=1}^{j_{max}} H(i,j)}{i_{max} \times j_{max}} \tag{30}$$

For the real case their normalized entropy is listed in the TABLE I:

The Fig. 7 shows a simulation in the environment that 50 nodes was deployed in a field 50×50, for each $\bar{H}$ the weight w is gotten by running for 100 time slots, when the movement-pattern is not obvious($\bar{H}$ is close to 1) the performance is almost the same as the LOCALE, when the movement-pattern increase($\bar{H}$ is coming to 0), however, the performance of our EMPAC becoming better and better. TABLE 1 show the normalized entropies of the five traces, from the table we can see that when the entropies are small corresponding to obvious movement-patterns the performances of our EMPAC

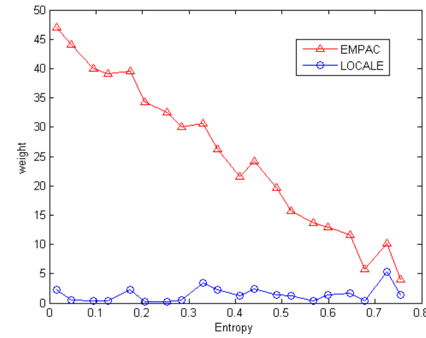

Fig. 8. The real world arithmetic comparison between EMPAC and LOCALE after 500 time slots

is almost $10\times$ better that the LOCALE. So our algorithm is more suitable for the real world case.

IV. Conclusion

In this paper, we introduced the EMPAC, a Exploiting Movement-Pattern Localization algorithm designed for the networks with different movement-pattern. We have show that with the consideration of the node's neighbor's location estimation and collaborative refine their location estimation, the location accuracy is $5\times$ better than just collaborative refine location information in the LOCALE.

The EMPAC's significance is that it regards different nodes with different attitudes based on their movement-patterns, the nodes with obvious movement-patterns could help those without refine their location estimations. This hypothesis is more suitable for the real world case.

References

[1] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee and Song Chong, *On the Levy-walk Nature of Human Mobility* In Proceedings of INFOCOM 2008, Arizona, USA, April 2008.

[2] Kyunghan lee, Seongik Hong, Seong Joon Kim, Injong Rhee and Song Chong, *SLAW: A Mobility Model for Human Walks*. In Proceedings of INFOCOM 2008, Rio de Janeiro, Brazil, April 2009.

[3] Zhang, P. and Martonosi, M. *Locale: Collaborative localization estimation for sparse mobile sensor networks*, Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on, pages 195–206, 2008.

[4] R. C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty, *The International Journal of Robotics Tesearch*, 5(4):56-68, 1986.

[5] Dragos Niculescu and Badri Nath. *Ad Hoc Positioning System(APS)Using AoA*, IEEE InfoCom 2003.

[6] B. H. Wellenhoff, H. Lichtenegger and J. Collins. *Global Positioning System: Theory and Practice*, Fourth Edition, Springer Verlag. 1997.

[7] Paramvir Bahl and Venkata N. Padmanabhan. *RADAR: An In-Building RF-Based User Location and Tracking System. IEEE InfoCom 2000, march 2000.*

[8] Neal Patwari and Alfred O. Hero III. *Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks*, Workship on Wireless Sensor Networks and Applications, September 2003.

[9] Andreas Savvides, Chih-Chieh Han, Mani B. Strivastava, *Dynamic fine-grained localization in Ad Hoc networks of sensors*, MobiCom 2001.

[10] STMicroelectronics. LIS3L02AQ DataSheet. *http://www.st.com/*, Nov. 2004.

[11] PNI Corporation. MicroMag3 Datasheet. *https://www.pnicorp.com/*, 2007.

[12] F. Ekman, A, Keränen, J. Karvo, and J. Ott, *Working day movement model*, In MobilityModels '08: Perceeding of the 1st ACM SIGMOBILE workshop on Mobility models, page 3340, 2008.
[13] W. Hsu, T. Spyropoulos, K. Psounis, and A, Helmy, *Modeling time-variant user mobility in wireless mobile networks*, In Prodeedings of INFOCOM 2007, Anchorage, AL, may 2007.
[14] M. Kim and D.Kotz, *Periodic properties of user mobility and accesspoint popularity*, Personal Ubiquitous Computing, 11(6):465-479, 2007.
[15] B. Kusy, G. Balogh, A. Ledeczi, and M. M. J. Sallai. inTrack: *High Precision of Mobile Sensor Nodes.* In 4th European Workshop on Wireless Sensor Networks(EWSN 2007), January 2007.
[16] W. E. Mantzel, C. Hyeokho, and R. G. Baraniuk. *Distributed Camera Network Localization*. In Signals, Systems and Computers, 2004.