Reconstructing Spatial Distributions from Anonymized Locations

James Horey [1], Stephanie Forrest [2], Michael Groat [3]

Computational Sciences & Engineering, Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
[1]horeyjl@ornl.gov

Dept. of Computer Science, University of New Mexico
Albuquerque, New Mexico, USA
[2]forrest@cs.unm.edu
[3]mgroat@cs.unm.edu

***Abstract*— Devices such as mobile phones, tablets, and sensors are often equipped with GPS that accurately report a person's location. Combined with wireless communication, these devices enable a wide range of new social tools and applications. These same qualities, however, leave location-aware applications vulnerable to privacy violations. This paper introduces the Negative Quad Tree, a privacy protection method for location aware applications. The method is broadly applicable to applications that use spatial density information, such as social applications that measure the popularity of social venues. The method employs a simple anonymization algorithm running on mobile devices, and a more complex reconstruction algorithm on a central server. This strategy is well suited to low-powered mobile devices. The paper analyzes the accuracy of the reconstruction method in a variety of simulated and real-world settings and demonstrates that the method is accurate enough to be used in many real-world scenarios.**

I. Introduction

With the proliferation of social networking applications, mobile devices, and urban sensor networks [1], location sharing has become a common online activity. Many mobile devices contain location sensors that can report a person's position with a high degree of accuracy. Social networking sites take advantage of this location information for a variety of applications. Examples include visualizing social hotspots [1], identifying traffic congestion [2], and informing friends of one's current location [2]. Although these applications provide many benefits, users still express strong privacy concerns [3].

Typically in these applications, mobile devices record the user's location, and transmit it to a central server. The server then aggregates locations in an application-specific manner. Many privacy and security issues can arise during the application lifecycle. Even when individual users are willing to reveal sensitive information, archiving such data may lead to inadvertent privacy breaches. One method to protect the user from privacy breaeches is to remove all unique identifiers from the location information. This can be accomplished by the user's mobile device before transmission.

[1]citysense.com
[2]www.google.com/latitude

Instead of anonymizing the user, an alternative approach obsfucates the location data. By obsfucating location data intelligently, individual users' privacy can be preserved without sacrificing the ability to authenticate users. A popular instantiation of this approach is *spatial cloaking* in which only a coarse view of the user's location is reported [6]. This technique can be combined with *k-anonymity* [7], where a sufficient large location area is reported to ensure that at least k individuals are co-located, making it difficult to know which one is the actual user. These techniques, however, still reveal the approximate location of a particular user. Also, the locations can often be correlated with other users if a history is stored. Decreasing spatial resolution can increase the k-anonymity, but may simultaneously harm the usability of the application. Yet another technique encrypts the location so data cannot be reconstructed if intercepted. Although this protects from unwanted snooping, the data must still be decrypted at the application server, which presents opportunities for malicious insiders or external parties to gain this information.

In our approach users report locations where they are not found. This process, called *negation*, enables users to participate anonymously in many location-based applications. Specifically, we target applications in which users are interested in *aggregated* location information. Using our technique, the negated locations can be reconstructed to compute the overall spatial distribution via a modified Negative Survey [8]. However, location-based services that report back specific locations of nearby services would need to use an alternative approach, such as one based on private information retrieval [9].

We introduce the Negative Quad Tree (NQT), an extension of the Negative Survey for reconstructing geographic density. Unlike location-cloaking, adversaries cannot even approximate where a particular user is located. In addition, adversaries cannot differentiate between multiple possible locations for a particular user. In the remainder of the paper we give details of the algorithm (Section II), and evaluate the algorithm, theoretically and via simulation, under various conditions (Section III). We show that the algorithm can anonymize data adequately while accurately reconstructing important informa-

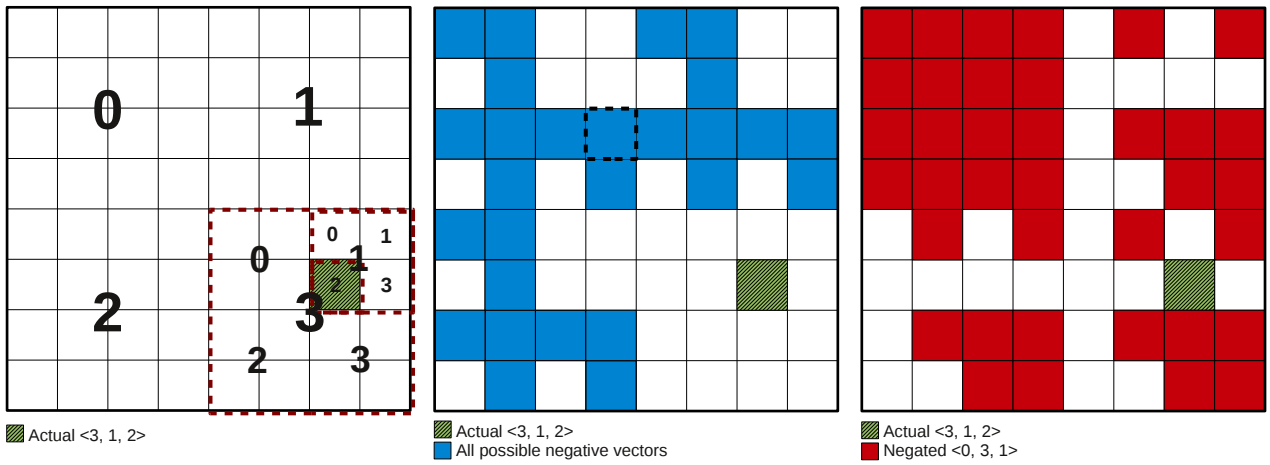

Fig. 1. The two-dimensional area is recursively divided into four quadrants. A location is encoded as a series of values identifying the quadrant starting with the upper-most set of quadrants. Given a location (green), the algorithm selects at random a negated vector (blue). Once a negative vector is selected (highlighted), the algorithm is able to exclude many locations (red).

tion. We also discuss potential vulnerabilities of our approach and demonstrate how it can defeat common correlation-based attacks (Section IV). Finally, we discuss related work (Section V) and offer a brief conclusion (Section VI).

II. Computational Model

The Negative Quad Tree algorithm consists of two phases. First, all locations are anonymized locally at the source via a mobile device. The anonymization process is designed to be simple and efficient. Next the anonymized data are collected at an application server. The server then reconstructs the spatial distribution of all the users. Finally, the reconstructed spatial distribution is transmitted back to users.

A. Anonymizing Location Data

The first step in the NQT algorithm is anonymization of the location data. Most GPS devices represent location as a pair of latitude and longitude values. This is converted to a quad tree format [10]. In a quad tree, the area of interest is divided into four rectangular quadrants. Each of these quadrants are numbered (0 - 3). Within each quadrant, the area is divided into sub-quadrants. This division repeats recursively until the desired resolution is reached. A location is encoded by recording all the quadrant values. With 5 hierarchical divisions (or *levels*), the quad tree will contain 1024 grid cells, sufficient for describing many metropolitan-sized areas (with $1km^2$ grid cells).

Algorithm 1 Negate(l)

Require: location l
  for $i = 0$ to $levels$ **do**
    $n_i \leftarrow random(\{0, 1, 2, 3\} - \{l_i\})$
  end for
  return n

For each element in the quad tree vector, the algorithm selects one of the other remaining quadrant values uniformly at random (Algorithm 1). For example, if the first vector value is 1, the algorithm chooses 0, 2, or 3. After completing this process for each vector value, the user is left with a *negative* vector.

Figure 1 illustrates this process for the location vector $< 3, 1, 2 >$. Once a negative vector is selected, the reconstruction algorithm then eliminates all locations represented by the negative vector. For example, if $< 0, 3, 1 >$ is chosen as the negative vector, the algorithm can eliminate all locations in quadrant 0. In the remaining quadrants, the algorithm can eliminate all locations in sub-quadrant 3. Within those remaining, the algorithm can finally eliminate sub-quadrant 1. After eliminating all these locations, there are still many remaining locations. The job of the reconstruction process is to estimate the number of samples found in the remaining locations.

B. Reconstructing Spatial Distributions

Algorithm 2 Estimate(l, r, e)

Require: location l, reported values r, current estimates e $\{\hat{l}$: negative vectors of l,
  c: contributions from other possible locations,
  $Pr_{m \to n}$: probability of m generating neg. vector $n\}$
  $s \leftarrow 0$
  for $n \in \{\hat{l}\}$ **do**
    $c \leftarrow 0$
    for $m \in \{\hat{n}\}$ **do**
      if $m \neq l$ **then**
        $c \leftarrow c + (Pr_{m \to n}) \times (e_m)$
      end if
    end for
    $s \leftarrow s + r_n - c$
  end for
  return s

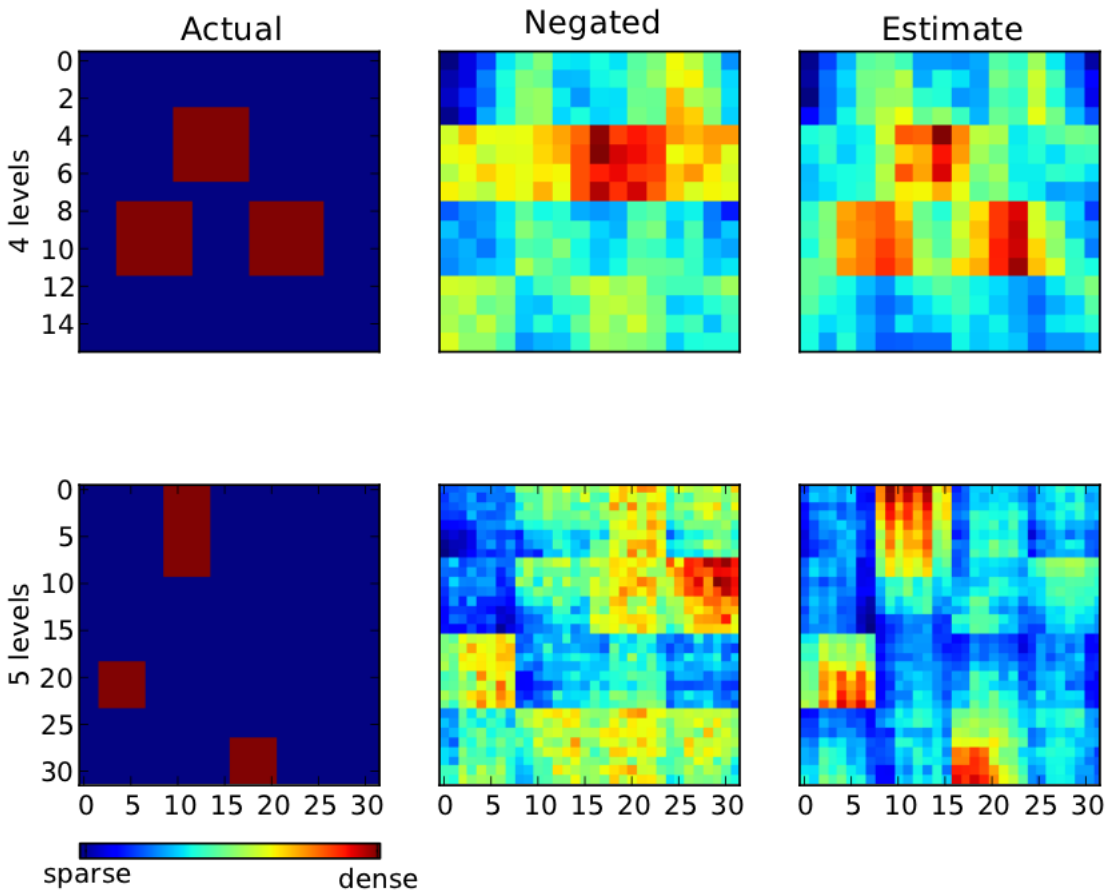

Fig. 2. Examples of the negation and reconstruction process for grids with 4 and 5 hierarchical levels using $128,000$ samples. Red indicates denser areas, while blue indicates sparse areas. The first column with actual data contains a few densely populated areas. The second column with the negated data obsfucates the data. The third column displays the reconstructed data.

Once all the negative vectors are collected, the NQT algorithm reconstructs the spatial distribution using a probabilistic approach. In order to estimate the number of samples for a location, the algorithm relies on an estimate for the other remaining locations. Consequently, the reconstruction process can be expressed as a linear system of equations. Each grid cell is modeled as a variable in the system. For a given grid cell, the coefficients of the equation can be calculated using Algorithm 2. This algorithm can be broken down into two steps. First, note that for a particular location l, there exists a select set of negative cells ($\hat{l}$) that may belong in l. By simply summing each value from $\hat{l}$, we can derive an upper estimate for l. However, there are many additional locations besides l that the negative cells may also belong to. In the second step, we simply remove those contributions.

We solve this system using the Apache Commons Java matrix library. The reconstruction process is relatively fast even for 5 quad levels (< 17 sec.) using a typical workstation (4 2.6 ghz Intel processors with 4 GB of RAM). Solving problems of the form $Ax = b$ is typically accomplished via LU factorization, an $O(n^3)$ process. For applications that require more than 5 quad levels, directly solving the linear system could be prohibitive. In that case, alternative approaches to solving the system may be necessary (e.g. Gauss-Seidel). However, for applications that take place in a typical metropolitan areas, $4 - 5$ quad levels should be sufficient.

III. Evaluation

Levels	*Negated Accuracy*	*Estimate Accuracy*
2	-0.356 $\pm$ 0.302	0.995 $\pm$ 0.302
3	-0.171 $\pm$ 0.231	0.874 $\pm$ 0.231
4	-0.093 $\pm$ 0.207	0.705 $\pm$ 0.207
5	-0.069 $\pm$ 0.059	0.518 $\pm$ 0.059

TABLE I

Average correlation values and standard deviations of negated and reconstructed histograms as the number of levels is increased from 2 to 5 using $128,000$ samples. Values close to -1.0 or 1.0 indicate a strong correlation.

We illustrate the results of negating and reconstructing the spatial distribution in two settings (Figure 2). In the top figure, we employ 4 hierarchical levels. Three large areas are more densely populated than the rest of the area (blue indicates low density, red indicates high density). In the bottom-most

figure, we employ 5 hierarchical levels, and densely populate four areas. Both settings use $128,000$ samples. The middle figures visualize the distribution of negated vectors, showing that they do not reveal much information about the actual distribution. Table I reports the Pearson correlation. Because negative vectors exclude large portions of the two-dimensional area, the Pearson correlation is better than 0. Overall, the reconstructed distributions identify salient features of the original distribution.

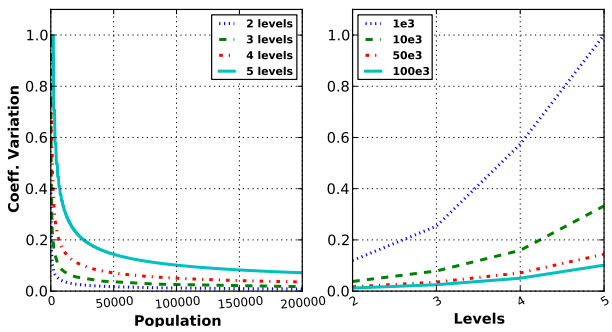

Fig. 3. Variance of the reconstruction as sample size and number of quad tree levels are increased.

To evaluate the efficacy of the reconstruction process, we model the coefficient of variation. We model the negation process as a process that uniformly distributes a value across all the remaining cells (excluding the original). This assumption allows us to model the reconstruction as a binomial process. The variance is then modeled using the standard coeffient of variation equation for the binomial distribution (sampling size n and l hierarchical levels):

$$v = \sqrt{\frac{(3^l * \frac{n}{4^l})(\frac{1}{3^l})(1 - \frac{1}{3^l})}{(3^l * \frac{n}{4^l})(\frac{1}{3^l})}} \tag{1}$$

Results are illustrated in Figure 3. As the sampling population increases, the variance quickly decreases and then levels off. Although variance is not a direct measure of reconstruction error, as variance decreases, the overall error is expected to decrease. As illustrated, however, with 5 levels, the variance does not reach 0 even with a high sampling population. As the number of levels is increased, the variance increases proportionally. With a sample size of 1000, the variance approaches 1 (at 5 levels), indicating that the reconstruction process will do poorly at those extremes.

We confirmed these modeling results using a set of simulated and actual population data. We simulated population data using three spatial distributions. First we simulated a uniform distribution with minor random variations in which users were spread uniformly throughout an area. We also simulated a *patchy* distribution, in which large groups of users were located in small patches. Finally, we simulated a *dense* distribution, in which most of the users reside in a few grid cells. We ran all experiments against these three distributions. For each experimental setup, we ran the experiment 10 times and averaged the accuracy values.

To measure accuracy, we compared the reconstructed histogram to the original histogram using the Pearson correlation coefficient (i.e. R value). The coefficient is calculated as follows:

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}} \tag{2}$$

where X refers to the original data and Y the reconstructed data. Given two similar histograms, the function will output a value close to -1 or $+1$ (indicating a strong negative or positive correlation between X and Y). Dissimilar histograms will output a value close to 0 (indicating no correlation).

In the first experiment, we evaluated the effects of increasing the spatial resolution as the number of samples varied (1000, 16000, 64000, and 128000). The accuracy steadily decreased with the number of levels (Figure 4) from a high of 1.0 to values between 0.4 and 0.6. This was true across all sample sizes and distributions. The uniform distribution, however, was more volatile and the reconstruction was easier with large sample sizes. We found that the patchy distribution performed better than the dense distribution. We should note, however, that even with 5 levels, the accuracy was still relatively high (> 0.4), indicating that our reconstruction method can work in high resolution scenarios.

We also evaluated the effects of increasing the number of samples on the accuracy as we varied the number of hierarchical levels (2, 3, 4, and 5). We observed that accuracy quickly increases and then steadily converges for most configurations (Figure 5). We also observed differences in the distributions more readily; distributions containing fewer dense areas performed better. However, the uniform distribution usually outperformed the other distributions after a sufficient number of samples (> 60000). Finally, we observed that because accuracy only increased slowly after a large number of samples (if at all); small fluctuations in sample size did not greatly affect the results. This can be both good (if there are fewer samples) or bad (since we cannot drastically increase the accuracy with more samples). However, for many social applications tens of thousands of participants is a realistic and sufficient figure.

In addition to simulation data, we evaluated the reconstruction technique on real world data. We downloaded geo-tagged images over a 1 year period from the photo-sharing site Flickr for the city of Barcelona, Spain. We obtained approximately $40,000$ points centered around the area near *Port de Barcelona* representing approximately a $25km^2$ area. We retroactively anonymized the data using 5 hierarchical levels. As Figure 6 illustrates, the reconstruction represents the popular area near the center and a few smaller popular areas surrounding the center. However, the center area is enlarged, and there are many false positives near the edges of the gridded area. The overall accuracy was approximately 0.59 indicating that the algorithm has difficulty capturing fine details. Refining our technique against additional real-world data is a subject of

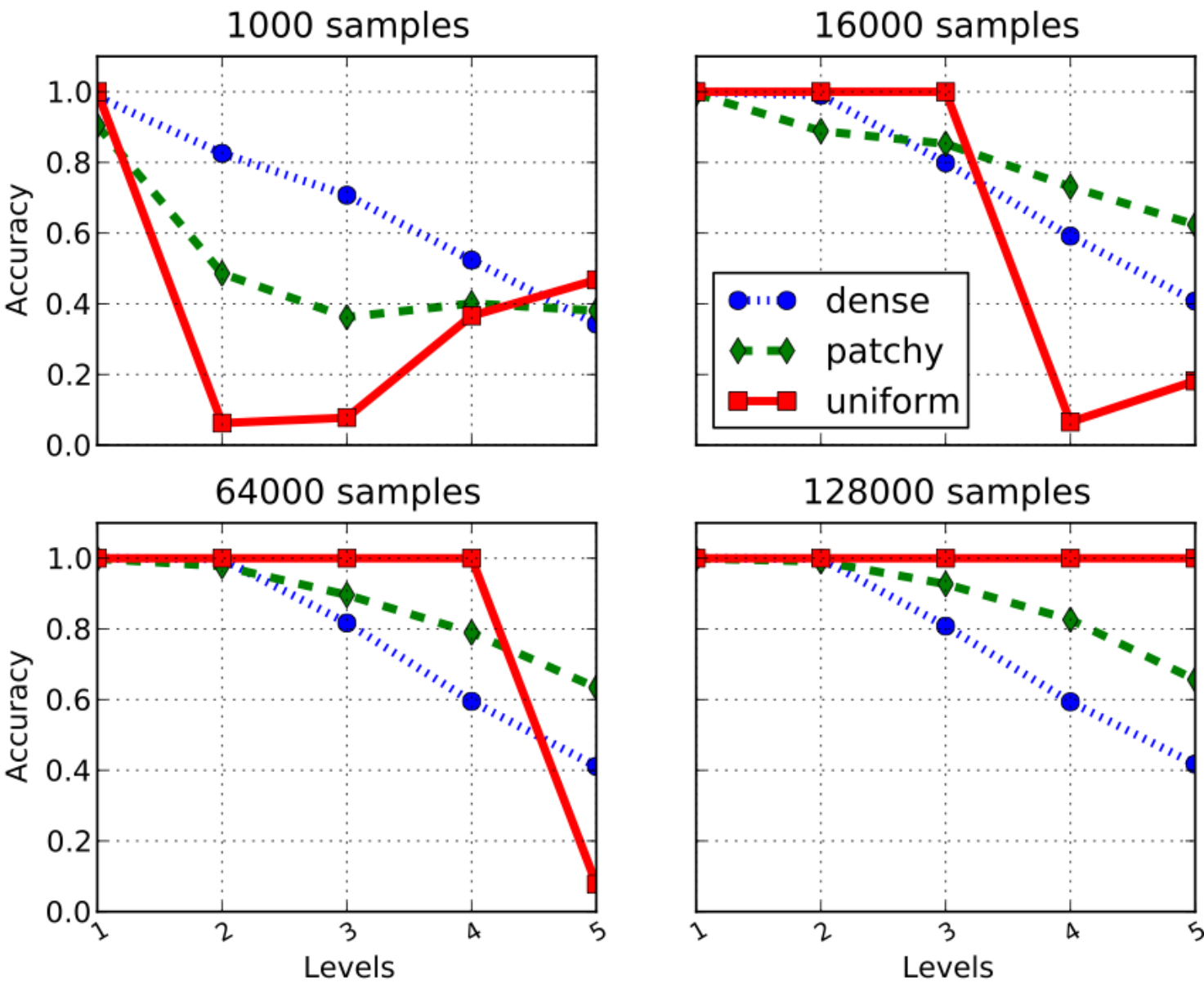

Fig. 4. Accuracy of the reconstruction as the number of hierarchical levels are increased from 1 to 5 using 1000, 16,000, 64,000, and 128,000 samples. The accuracy decreases with more levels for nearly all sample sizes.

future work.

The results of these experiments suggest that the NQT reconstruction is accurate for a wide array of sampling and resolution scenarios. In general, lower spatial resolutions and higher sampling rates increase overall accuracy as one would expect. This is true for many different types of spatial distributions. The NQT reconstruction works especially well on large, dense areas. Of course, applicability depends on the nature of the application requirements. For example, applications that are designed to pick out social hotspots will perform well, because they are insensitive to small inaccuracies.

IV. Correlation-Based Attacks

Many applications rely on collecting location data periodically. Assuming that users move slowly (e.g. walking), an adversary may be able to perform a *correlation-based attack* to guess a user's location. To execute this attack, the adversary must first gain access to a history of the user's negative vectors. Given two negative vectors, one at time i and another at time $i+1$, the adversary could generate all possible locations for each negative vector and then perform a pair-wise comparison between the two sets. Any pair of locations that is geographically close to each other has a strong likelihood of being the actual user location. The adversary can then

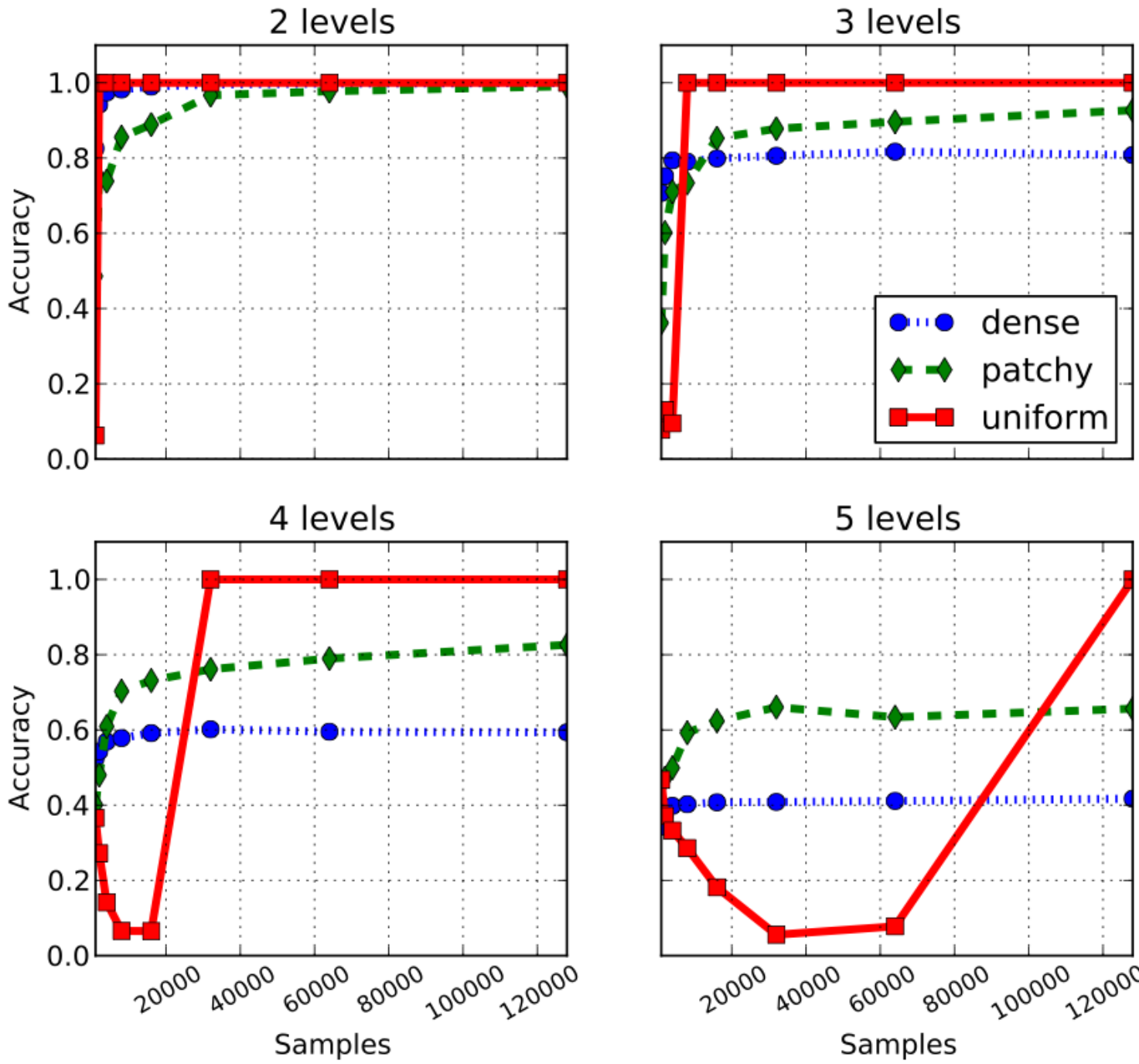

Fig. 5. The accuracy of the reconstruction as the number of samples are increased. The accuracy usually increases quickly with more samples, and eventually levels off.

further decrease the number of possible pairs by continuing this process.

To address this attack, the Negative Quad Tree can be slightly modified to *minimize* differences between subsequent negative vectors. First, a very short history of previous negative vectors is stored locally on the user's device. If a user moves to a nearby grid cell, the algorithm does not generate a completely new negative vector. Instead, the algorithm attempts to re-use as many elements from one of the previous negative vectors. Assuming that the user has not moved very far, most (if not all) of the negive vector elements can be re-used. Implementing this optimization greatly reduces the potential for a correlation-based attack (Figure 7). In the non-optimized case, the number of potential locations decreases quickly for scenarios in which the user is moving along both a straight line and a random walk. However, for the optimized algorithm,

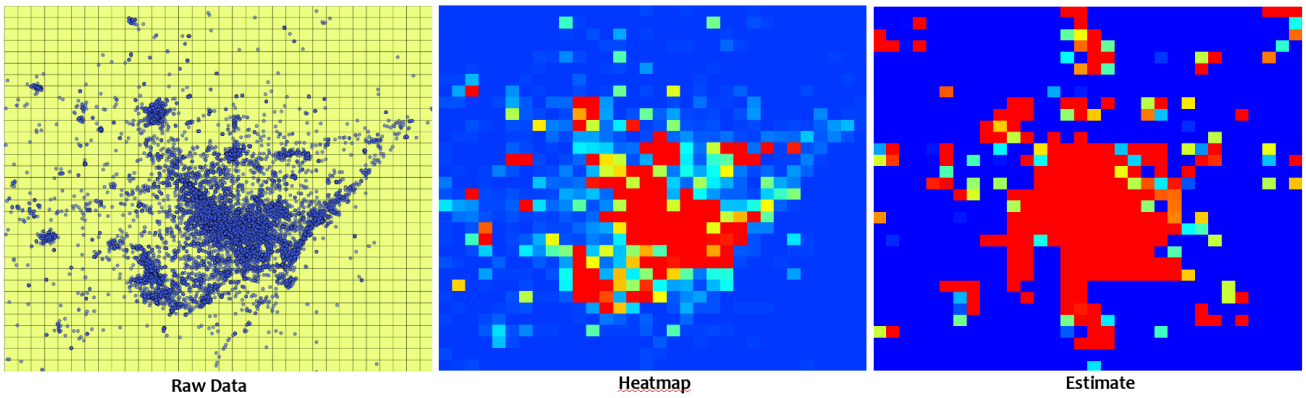

Fig. 6. Photo data obtained from Flickr of Barcelona, Spain over a one year period with $40,000$ samples. The right-most figure is the reconstruction of the original data using the NQT method.

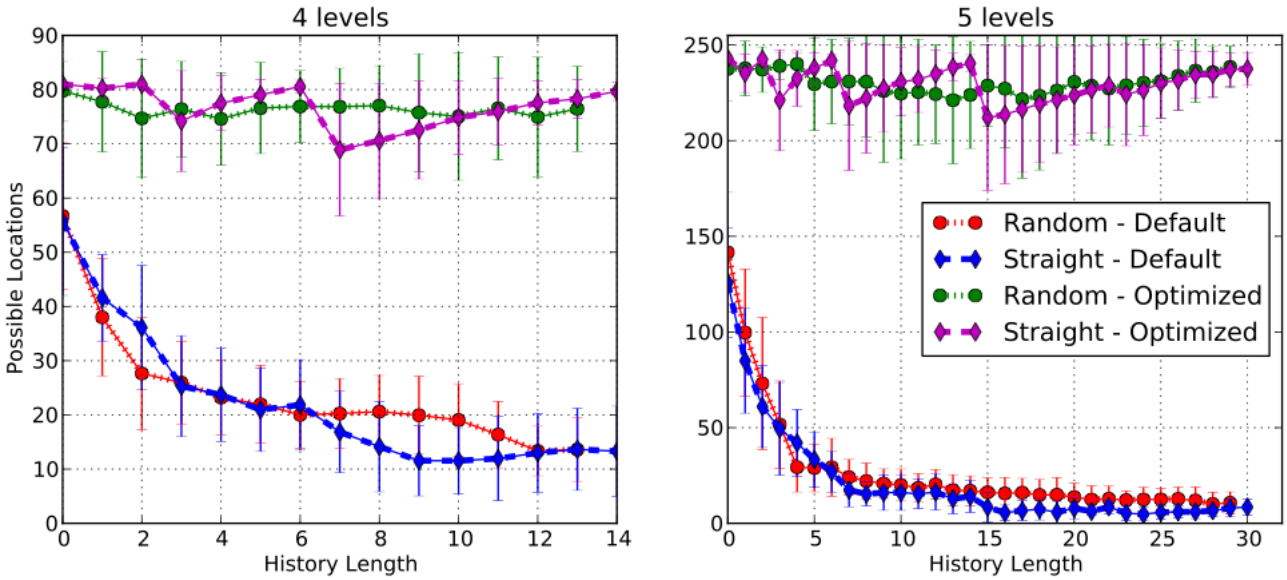

Fig. 7. The number of locations where the user may be located decreases quickly as the history size is increased. The difference-minimizing optimization maintains a high number of possible locations and successfully defends against this attack.

the number of potential locations stays very high regardless of the history size. This makes it substantially more difficult to guess the user's location.

A. Negative Survey

The NQT algorithm extends earlier work on the negative survey. In the original work, users report a single value by choosing randomly from a set of discrete categories excluding their own. In principle, this method could be applied to spatial density estimation by treating each location as a discrete category. However, the original negative survey has difficulty handling large numbers of categories, making the technique infeasible for spatial applications (the number of locations grows as 3^{levels}). More recently there has been work to improve the reconstruction accuracy of the negative survey by assuming Gaussian priors [12]. This improved accuracy enables the authors to consider spatial reconstruction. The NQT, in contrast, does not make any a-priori assumptions regarding the spatial distribution.

V. Related Work

Recent work on participatory, urban sensing re-enforces the need for privacy protection [13]–[15]. In these schemes, centralized servers sanitize private data in application specific ways. The NQT algorithm, in contrast, anonymizes the location data at the point of collection, minimizing the risk of confidentiality loss. When the actual location data is transferred, cryptographic techniques are often used to protect data transmit. Recent work shows that it is possible to use encryption techniques on low power devices [16], [17]. Cryptographic techniques emphasize data security, but because the data must be decrypted to be used they don't provide full data privacy.

AnonySense [18] is a generic privacy framework designed

for use with personal devices within urban areas. Users employ a tasking language to specify the type of data to be collected from these devices. The system then anonymizes user information using a MIX network [4]. Unlike our work, AnonySense is concerned with anonymizing the *source* of the data, rather than the data itself. Due to its use of a MIX network, the system requires a more complex anonymization and authentication scheme in which the user must implicitly trust certain system services (e.g. the mixing components). Our technique is relatively simple and requires fewer trusted components.

Our technique has the same goal as data perturbation methods [19]–[21], where pseudo-random noise is added to a set of data to obfuscate it. However, these techniques are often designed to operate in a continuous domain, while many location-based applications draw data from a discrete domain. Our work also shares many goals with privacy-preserving data aggregation techniques [22]–[24], in which sensor nodes transmit anonymized data. The anonymized data are aggregated in such a way that aggregate functions can be easily computed. Our work generalizes these techniques since histograms can be used to compute various aggregate values.

Randomized response techniques (RRTs) [19] are a survey method designed to estimate the proportion of a population that belongs to a particular group while protecting the privacy of individuals participating in the survey. It does this by offering surveyors multiple questions (only one of which is sensitive) in lieu of a single question. Individuals randomly select one of the questions to answer. Individuals give a yes or no answer to one of these questions, but do not reveal which question was answered. In this way, the results of the survey combined with the characteristics of the randomizing device provides enough information to reconstruct the proportion of population members in each group.

Another important class of applications that require privacy guarantees are online streaming applications. The primary privacy goal in such applications is to ensure that the internal state can be updated in a streaming manner without storing sensitive data. Work by Dwork et al [25] addresses hese issues using a variety of methods including randomized response. The primary difference between these applications and the ones we address is the assumption that all the data used for reconstruction are readily available.

Our technique is inspired by negative databases [26], [27]. A negative database stores a compressed form of the data complement instead of the actual data. The subsequent database can be queried for element membership in polynomial time. However, reconstructing the original database is difficult and formally NP-Hard. Other operations over the negative database are possible and range in computational complexity [28].

VI. Conclusion and Future Work

The ability to collect location data has created many interesting and useful applications. These applications range from providing social services to providing key information on traffic conditions. Although useful, location-aware applications also have the potential to be abused. Software that helps users find nearby friends can be compromised and reveal private information. Similarly, applications may reveal information about users that unwittingly disclose private information. To address these issues, we developed and evaluated the Negative Quad Tree, a privacy-preserving method that addresses the construction of spatial densities using anonymous location data. We evaluated the algorithm under a variety of scenarios and demonstrated that it can be used in many real-world settings.

Although the Negative Quad Tree addresses a wide array of applications, the reconstruction is not perfect and may not be suitable for all applications. We are focusing our future work on improving the accuracy of the reconstruction process for more stringent applications (including those that require higher spatial resolutions). One approach we are exploring is combining location cloaking with the NQT. Assuming that users are comfortable broadly describing their location (i.e. "south Knoxville"), we can employ the NQT in a smaller area. In addition, we are also exploring the use of multi-dimensional negative surveys in which multiple values must be hidden (including location). We are confident that these extensions will enable the use of the negative surveying techniques in a variety of future applications.

As more devices become location-aware, the need for privacy protection can only increase. Privacy protection will come in multiple forms including legal constructs, cryptographic frameworks, and application specific algorithms. Our methods fall into application specific algorithms, because we address location privacy within the context of certain application scenarios. As location-sensitive technology matures, we expect that privacy requirements will be addressed through a combination of these approaches. By incorporating the Negative Quad Tree algorithm into users' devices, users will have an additional tool by which to control their privacy.

VII. Acknowledgements

JH thanks Alison Boyer and Blair Sullivanfor their feedback and suggestions. This work was funded by Oak Ridge National Laboratory, National Science Foundation (grants CCF 0621900, CCR-0331580, SHF-0905236), Air Force Office of Scientific Research MURI grant FA9550-07-1-0532, and the Santa Fe Institute. This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

[1] R. Murty, G. Mainland, I. Rose, A. R. Chowdhury, A. Gosain, J. Bers, and M. Welsh, "Citysense: A vision for an urban-scale wireless networking testbed," in *IEEE International Conference on Technologies for Homeland Security*, 2008.

[2] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Maadden, "Cartel: A distributed mobile sensor computing system," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[3] E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Sprintfield, L. Cranor, J. Hong, and N. Sadeh, "Empirical models of privacy in location sharing," in *International Conference on Ubiquitous Computing (UbiComp)*, 2010.

[4] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[5] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Communications of the ACM*, vol. 42, pp. 39–41, February 1999. [Online]. Available: http://doi.acm.org/10.1145/293411.293443

[6] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2003.

[7] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, fuzziness and Knowledge-Based Systems*, vol. 10, pp. 557–570, October 2002.

[8] J. Horey, M. Groat, S. Forrest, and F. Esponda, "Anonymous data collection in sensor networks," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous)*, 2007.

[9] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *ACM SIGMOD international conference on Management of data*, 2008.

[10] R. Finkel and J. Bentley, "Quad trees: A data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, no. 1, pp. 1–9, 1974.

[11] F. Esponda and V. M. Guerrero, "Surveys with negative questions for sensitive items," *Statistics & Probability Letters*, vol. 79, no. 24, pp. 2456–2461, 2009.

[12] H. Xie, L. Kulik, and E. Tanin, "Privacy-aware collection of aggregate spatial data," *Journal of Data & Knowledge Engineering*, vol. 70, pp. 576–595, June 2011.

[13] S. Reddy, G. Chen, B. Fulkerson, S. J. Kim, U. Park, N. Yau, J. Cho, M. Hansen, and J. Heidemann, "Sensor-internet share and search: Enabling collaboration of citizen scientists," in *Workshop for Data Sharing and Interoperability - IPSN*, 2007.

[14] M. Srivastava, M. Hansen, J. Burke, A. Parker, S. Reddy, T. Schmid, K. Chang, G. Saurabh, M. Allman, V. Paxson, and D. Estrin, "Network system challenges in selective sharing and verification for personal, social, and urban-scale sensing applications," in *Workshop on Hot Topics in Networks (HotNets)*, 2006.

[15] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, 2009.

[16] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks," *International Journal of Security and Networks*.

[17] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[18] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: An architecture for privacy-aware urban sensing."

[19] S. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[20] R. Agrawal and R. Srikant, "Privacy-preserving data mining," 2000.

[21] S. Zhang, J. Ford, and F. Makedon, "Deriving private information from randomly perturbed ratings," in *Siam Conference on Data Mining*, 2006.

[22] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "Pda: Privacy-preserving data aggregation in wireless sensor networks," in *IEEE Conference on Computer Communications (InfoCom)*, 2007.

[23] W. He, X. Liu, H. Nguyen, and K. Nahrstedt, "A cluster-based protocol to enforce integrity and preserve privacy in data aggregation," in *International Workshop on Cyber-Physical Systems (WCPS)*, 2009.

[24] M. M. Groat, W. He, and S. Forrest, "Kipda: k-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *IEEE Conference on Computer Communications (InfoCom)*, 2011.

[25] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin, "Pan-private streaming algorithms," in *ACM Symposium on Theory of Computing (STOC)*, 2010.

[26] F. Esponda, S. Forrest, and P. Helman, "Negative representations of information," *International Journal of Information Security*, vol. 8, no. 5, 2009, doi:10.1007/s10207-009-0078-1.

[27] F. Esponda, E. S. Ackley, P. Helman, H. Jia, and S. Forrest, "Protecting data privacy through hard-to-reverse negative databases," in *Information Security Conference*, 2006.

[28] F. Esponda, E. D. Trias, S. Forrest, and E. S. Ackley, "A relational algebra for negative databases," University of New Mexico, Tech. Rep., 2007.