Abstract—The popularity of GPS-enabled smartphones enables a wide variety of new location-based or location-aware services and applications. However, the GPS module in a smartphone produces inaccurate position information and incurs high energy consumption, which inhibits the wide use of location-aware applications. To address this, we propose a social-aided cooperative localization scheme, which is capable of improving positioning accuracy and achieving low energy consumption. Specifically, our scheme enhances positioning accuracy by fusing the GPS positions of multiple co-located smartphones in a social network, or by neighborhood-based weighted least-squares estimation when relative distances between smartphones are available. The energy efficiency is achieved by sharing location information among co-located users so that only a small set of smartphones need to turn on GPS receivers at any time. To validate our proposed approach, we conduct experiments on the Google Android platform. Experimental results show that our proposed cooperative localization scheme can achieve sufficient performance gains in both indoor and outdoor environments.

I. INTRODUCTION

Most new models of smartphones have built-in Global Positioning System (GPS) receivers. The GPS onboard enables a host of location-aware applications. According to a study published by the Pew Internet and American Life Project [1], more than 55% of smartphone owners use their phones to find directions, recommendations, or other information related to their present locations. In addition, geo-social “check in” services such as Foursquare or Gowalla are very popular among young adults [1]. New digital cameras or smartphones are equipped with geo-tagging features [2], making it easy to group photos by location or track the user’s footprint.

GPS is a ranging-based localization system, with a worst-case pseudorange accuracy of 7.8 meters at a 95% confidence level [3]. However, obtaining the GPS position information incurs a high cost; the whole process includes many complex calculations, e.g., correlation, demodulation, tracking, ranging and positioning. Moreover, satellite signals are hard to access especially in indoor and harsh environments due to the strong attenuation of the radio caused by building materials. The process of constantly searching and capturing the very weak beacon signal consumes a lot of power, and the estimated position is often inaccurate or even unavailable.

The rapid growth in people-centric mobile computing applications and location-based services has called for improved localization techniques. Energy-efficiency and accuracy are the two main objectives of such improvements. Authors in [4] [5] [6] have paid attention to tradeoff between energy and location accuracy. They try to use low power WiFi/GSM based schemes to lower the frequency of GPS startups, but at the expense of lower accuracy and update rates. Other approaches utilize dedicated devices for localization when GPS signal is unavailable. However, a lot of anchor nodes need to be placed at a very high density with known coordinates.

One compelling technique for improving the performance of localization without relying on the infrastructure is cooperative localization [7]. Cooperation among peer nodes at the physical layer can improve the communication capacity and coverage of wireless networks [8]. Recently, such a cooperation paradigm has been introduced for localization and navigation to improve the accuracy and reliability of positioning and circumvent the need for high-power infrastructure [9]. In this paradigm, it is assumed that devices can take intra- and inter-node ranging measurements in addition to measurements with respect to anchor nodes, since measurements with respect to anchor nodes only are insufficient for (accurate) positioning in harsh indoor/outdoor environments [10]. By exchanging the anchor node information and performing relative ranging between nodes, the position estimation for each node becomes possible or more accurate [11], [12].

However, existing cooperative localization techniques [8], [11], [12] require access to physical-layer processing units. But a commercial smartphone does not allow users to access its physical-layer processing units; and the GPS position of the smartphone is the only information accessible by a user/application. Therefore, the existing cooperative localization techniques [8], [11], [12] are not directly applicable to GPS-enabled smartphones.

In this paper, we extend the cooperative localization technique to a social network setting. Unlike conventional cooperative localization that utilizes physical-layer information fusion, our proposed social-aided cooperative localization (SCL) performs data fusion at the application layer when GPS positions of smartphones are already known. Application-layer fusion can achieve performance improvement at a lower cost. The rationale is that when a group of people in a common location all carry smartphones with GPS capability, the accuracy of localization can be significantly improved by fusing the GPS positions of the smartphones in this group. Theoretically, this performance gain is ascribed to the law of large numbers.
Further, once the refined position is obtained, in the next iteration of the algorithm, only one or two selected members in this group need to update their GPS positions; non-selected co-located members can use the locations of the selected members as their own; and GPS-active candidates can be selected in a round robin manner to balance power consumption among all smartphones. This can reduce the total power consumption dramatically. We want to emphasize that the driving force of cooperative localization is the fast-improving smartphone technology and people-centric pervasive social computing.

The rest of this paper is organized as follows. Section II discusses our system design. Section III describes our cooperative localization scheme for the case that co-location information is available. Section IV presents our cooperative localization scheme for the case that relative distances between smartphones in a neighborhood are known. Section V and Section VI present numerical results and experimental results, respectively. Section VII concludes the paper.

II. SYSTEM DESIGN

Fig. 1 illustrates the overall system setting and basic operations for social-aided cooperative localization. In the following subsections, we sketch an overview of the system architecture first, then describe some important steps in system operation: (i) Smartphones upload information to the server; (ii) Server performs clustering and position refinement; (iii) Server sends the refined position back to the smartphones.

A. Social Cooperative Scheme

To realize cooperative localization via social collaboration, we propose a scheme for the smartphone to collect data and use the refined results calculated by the server. The social cooperative scheme consists of the following five steps:

1) Initialization:
Each smartphone obtains its position by its own GPS receiver during the start-up period. Assume that there are WiFi access points near each smartphone and each smartphone can obtain the identification (ID) numbers of its nearby WiFi access points and the radio signal strengths (RSS) of the WiFi channels. Then, each smartphone reports the following to the server: its GPS position, IDs of its nearby WiFi access points and the corresponding RSS.

2) Clustering and Co-location Detection:
The server performs clustering [13], [14] to form groups of co-located smartphones, i.e., partition all the smartphones into groups according to their GPS positions and WiFi RSS.

3) Distance Estimation:
We estimate the relative distance between smartphones. If physical ranging is not available, the relative distance between any two smartphones can be calculated from the GPS positions of the two smartphones. If physical ranging (such as beaconing over a WiFi channel or using the acoustic signal for time-of-arrival estimation) is available, higher positioning accuracy can be achieved in Step 4.

4) Position Optimization:
The server invokes the position optimization algorithm (i.e., neighborhood-based weighted least-squares estimation algorithm) to refine the position of each user by utilizing users’ co-location information, the (coarse) GPS position information obtained in Step 1, and the relative distances obtained in Step 3. We develop two position refinement algorithms: one for the case without relative distances, and the other with relative distances.

5) Position Updating:
The server sends the refined position back to each smartphone; and each smartphone updates its position with the received value.

We assume that users are pedestrian and their movement is insignificant, compared with the location updating frequency. After finishing Step 5, the next iteration goes back to Step 1. In the new Step 1, only one or two selected members in this group need to update their GPS positions; non-selected co-located members can use the locations of the selected members as their own. We rotate members’ GPS duty periodically to balance power consumption among all smartphones in a group. Without activating GPS for all phones, this scheme can dramatically reduce total power consumption.

If one user uses a GPS-incapable phone, the initialization step in this scheme is different. In the initialization step, each GPS-incapable phone only needs to upload its WiFi RSS to the server for clustering and co-location detection. In the last step, the phone can simply use the position of a co-located collaborator as its own.

B. Use Cases

Our social-aided cooperative localization scheme can be applied to two scenarios. The first scenario is a smartphone being in an indoor environment where GPS signal is too weak to be detected. This smartphone is treated as a GPS-incapable phone; our social-aided cooperative localization scheme can leverage nearby outdoor smartphones to get a refined position for the indoor smartphone. The first step is that the indoor smartphone searches nearby outdoor GPS-enabled smartphone, and requests their cooperation. Then ranging
between any pair of the smartphones is conducted. The relative distances and GPS positions obtained are used by the position optimization algorithm, which produces a refined position for each smartphone including the indoor one. Even if relative distances are unavailable, our co-location based cooperative localization scheme can still produce a refined position for an indoor smartphone.

The second scenario is that when you go outside with your friends, you and your friends are in the same route or bus, and many of you use smartphones to take geo-tagging photos occasionally. Without cooperation, every time when you open your camera application, the smartphone starts up the GPS to perform localization. Even if your group is in the same place, anyone who wants to take photos needs to start up the GPS again, which slows down the response of the camera and drains the battery. Moreover, using one snapshot GPS measurement for localization is inaccurate. You may have experienced that some of your photos in your album show some places that you had never been before due to inaccurate GPS results. To improve positioning accuracy, you can just use the shared position, which is more accurate and power efficient than direct calculation by themselves.

III. COOPERATIVE LOCALIZATION UNDER CO-LOCATION

In this section, we focus on utilizing the co-location information to improve positioning accuracy.

A. Mathematical Modeling

We consider a social network consisting of \( m \) collaborators in \( \mathcal{R}^d \), where \( d \) is the coordinate dimension, i.e., \( d = 3 \) for ellipsoidal space; \( d = 2 \) for the cartesian space. Let \( \mathcal{N}_g = \{1, 2, \ldots, m\} \) denote the set of collaborators.

Assume the ground truth position of each collaborator is \( \mathbf{p}_i \), \( i \in \mathcal{N}_g \). With ellipsoidal coordinates, \( \mathbf{p}_i \) can be written as a form of latitudes (radians), E longitudes (radians) and heights (m), i.e., \( \mathbf{p}_i = (\text{lat}_i, \text{lon}_i, h_i)^T \). To simplify the process, we can change the ellipsoidal coordinates to the cartesian coordinate under the standard of Geodetic Reference System 1980 (GRS80) by function \( \mathbf{p}_i(x, y, z) = f_{\text{GRS80}}(\mathbf{p}_i(\text{lat}, \text{lon}, h)) \) as

\[
\begin{align*}
    v & = a \sqrt{1 - e^2 \sin^2(\text{lat})} \\
    x & = (v + h) \cos(\text{lat}) \cos(\text{lon}) \\
    y & = (v + h) \cos(\text{lat}) \sin(\text{lon}) \\
    z & = (v(1 - e) + h) \sin(\text{lat})
\end{align*}
\]

where \( a \) and \( e \) are the reference of ellipsoid major semi-axis and eccentricity squared parameters defined in GRS80.

For small-scale geographic space, we can focus on the 2D cartesian coordinate without the heights \( (h) \) information. By subtracting a pre-defined reference point \( \mathbf{p}_{ref} = (x_f, y_f)^T \), a local coordinate obtained by the GPS module in smartphone is \( \mathbf{p}_i = \mathbf{p}_i - \mathbf{p}_{ref} = (\tilde{x}_i, \tilde{y}_i)^T \), \( i \in \mathcal{N}_g \) for plane-coordinate. The estimation error is \( \mathbf{e}_i = |\mathbf{p}_i - \hat{\mathbf{p}}_i| \). Assuming that the position estimate is unbiased, \( \mathbf{e}_i \) follows a zero-mean Gaussian distribution as \( \mathbf{e}_i \sim \mathcal{N}(0, \Sigma_i) \), where \( \Sigma_i \) is the error covariance matrix and is assumed to be a diagonal matrix with diagonal entries of \( (\sigma_i^x)^2 \) and \( (\sigma_i^y)^2 \). Then, the position matrix of each collaborator can be written as \( \mathbf{P} = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \ldots, \hat{\mathbf{p}}_m] \in \mathcal{R}^{d \times m} \).

The problem of social-aided cooperative localization can be modeled as to refine the estimated positions \( (\hat{\mathbf{p}}_i) \) obtained by GPS. The additional information that we utilize to optimize the accuracy of GPS position are the co-location or relative distances \( (\mathbf{D}) \) between collaborators.

B. Co-location Detection and Clustering

In the step of grouping, we use a distributed clustering algorithm to form groups, i.e., partition all the smartphones into groups. Widely-used clustering algorithms include K-means, un-normalized spectral clustering, the G-cut algorithm, and the normalized cuts algorithm.

K-means algorithm, also known as (a.k.a.) Lloyd algorithm [13]. It is based on the nearest neighbor criterion and the centroid criterion. However, K-means algorithm cannot be used for grouping smartphones if the GPS positions of smartphones are not available due to significant signal attenuation in indoor environments, high rise building environments, or dense forest.

In the following three environments, namely, 1) indoor environments, 2) high rise building environments, and 3) dense forest, GPS signal is too weak to detect. We propose the following method to obtain distances or affinity measures between any pair of smartphones. Suppose the network under consideration consists of \( N \) smartphones with Wi-Fi communication capability. Each smartphone uses exponential backoff procedure to gain access to the Wi-Fi channel. When a smartphone (say, Node \( i \)) obtains access to the Wi-Fi channel, it sends a beacon signal. Any smartphone (say, Node \( j \) (\( j \neq i \))) that is able to detect the beacon signal, records the power \( P_{ij} \) of the received beacon, and \( P_{ij} \) will be regarded as an affinity measure between Node \( i \) and Node \( j \); for any node \( k \) that is not able to detect the beacon signal, we set \( P_{ik} = 0 \). Note that given a path loss model and transmission power \( P_i^{(t)} \), we can convert \( P_{ij} \) to distance \( d_{ij} \) between Node \( i \) and Node \( j \), up to a constant scaling factor. For example, assume the path loss is proportional to the \( n \)-th power of distance, i.e., \( P_i^{(t)} / P_j^{(t)} = \kappa \times d_{ij}^n \), where \( \kappa \) is a constant; then \( d_{ij} = \kappa^{-\frac{1}{n}} \times (P_i^{(t)} / P_j^{(t)})^{\frac{1}{n}} \).

Once each smartphone gets a chance to send a beacon signal, we can obtain an affinity matrix \( S \) (where \( S \in \mathcal{R}^{N \times N} \)) and the entry at the \( i \)-th row and the \( j \)-th column of \( S \) is \( P_{ij} \). We assume that the transmission power \( P_i^{(t)} \) is the same for all \( i \). In case that \( P_i^{(t)} \) are different for different \( i \), the affinity matrix \( S \) will not be symmetric; then we will use
\[ S_{sym} = S + S^T \] as the affinity matrix, which is symmetric. Un-normalized spectral clustering [14] and the normalized cuts algorithm [15] are able to group smartphones, given an affinity matrix.

C. Location Fusion under Co-location

If the relative distances between users are unknown, performance gains can be only achievable when the collaborators are close to each other.

The estimated position error for user \( i \) from GPS module is \( e_i \). As stated in Section III-A, \( e_i \) follows a zero-mean Gaussian distribution as \( e_i \sim N(0, \Sigma_i) \). The estimated position value \( \hat{p}_i \) follows a Gaussian distribution with mean \( p_i \) and the same covariance matrix, i.e., \( \hat{p}_i \sim N(\mathbf{p}_i, \Sigma_i) \). The probability density function of \( \hat{p}_i \) is given by

\[
f(\hat{p}_i) = \frac{1}{\sqrt{2\pi \det(\Sigma_i)}} \exp \left( -\frac{D^T \Sigma_i^{-1} D}{2} \right)
\]

where \( D = (\hat{p}_i - p_i) \), and \( \det(\Sigma_i) \) calculates the determinant of \( \Sigma_i \).

Co-location information of collaborators can be utilized due to the correlation between different estimated positions. For the co-location clusters \( C_1, \ldots, C_K \), the mixture of the position information of cluster \( C_k \) can be written as

\[
\hat{p}_k = \frac{1}{N_k} \sum_{i \in C_k} \gamma_i \hat{p}_i
\]

where \( \gamma_i \) is the weighting coefficient of initial location for users in cluster \( C_k \), and can be calculated by the historical position variance of user \( i \).

To illustrate the performance gains with regard to the maximum allowable distance, we focused on the location fusion of two users case with \( \hat{p}_{i,j} = \gamma_i \hat{p}_i + \gamma_j \hat{p}_j \). The probability density function of the mixed random variable \( \hat{p}_{i,j} \) is

\[
f(\hat{p}_{i,j}) = \gamma_i f(\hat{p}_i) + \gamma_j f(\hat{p}_j).
\]

If equal weighting method is used for information fusion, the coefficients are \( \gamma_i = \gamma_j = \frac{1}{2} \). The location estimation result \( \hat{p}_{i,j} \) still follows a Gaussian distribution as \( \hat{p}_{i,j} \sim N(\mathbf{p}_{i,j}/2, \Sigma_{i,j}) \), where \( \Sigma_{i,j} \) is a diagonal matrix with diagonal entries of \((\sigma_i^2)^2 + (\sigma_j^2)^2)/4\) and \((\sigma_i^2)^2 + (\sigma_j^2)^2)/4\).

The mean square error (MSE) is often used as a characteristic metric to illustrate the accuracy of the estimation result. The MSE of the estimation of \( \hat{p}_i \) is \( MSE_i = (\sigma_i^2)^2 + (\sigma_i^2)^2 \). Define \((\sigma_i^2)^2 = (\sigma_j^2)^2 = (\sigma_i^2)^2 + (\sigma_j^2)^2 \). The MSE of \( \hat{p}_{i,j} \) is given by

\[
MSE_{i,j} = E[\| \mathbf{p}_j - \hat{p}_{i,j} \|^2]
\]

\[
= E[\| \mathbf{p}_j - (\hat{p}_i + \hat{p}_j)/2 \|^2]
\]

\[
= \frac{1}{4} \| \mathbf{p}_j - \mathbf{p}_i \|^2 + \frac{1}{4} (\sigma_i^2)^2 + \frac{1}{4} (\sigma_j^2)^2
\]

where \( \| \mathbf{p}_j - \mathbf{p}_i \|^2 \) is the 2-norm of the distance difference, i.e., the biased value of the estimator. The MSE for the initial position estimation result is \( MSE_i = (\sigma_i^2)^2 \). The \( \hat{p}_{i,j} \) can be defined as the difference of the MSE value as

\[
\Delta MSE_{i,j} = \frac{1}{4} \| \mathbf{p}_j - \mathbf{p}_i \|^2 + \frac{1}{4} (\sigma_i^2)^2 + \frac{1}{4} (\sigma_j^2)^2 - (\sigma_i^2)^2
\]

\[
= \frac{1}{4} \| \mathbf{p}_j - \mathbf{p}_i \|^2 + \frac{1}{4} (\sigma_j^2)^2 - \frac{3}{4} \sigma_i^2 (\sigma_j^2)^2
\]

In order to achieve performance gains for user \( i \) when using the position of user \( j \) for information fusion, the condition \( \Delta MSE_{i,j} < 0 \) should be satisfied. Define the performance gain of user \( i \) using the position information from \( i \) and \( j \) as \( G_i(i,j) = -\Delta MSE_{i,j} \). The necessary condition can be shown as

\[
\| \mathbf{p}_j - \mathbf{p}_i \|^2 < 3(\sigma_i^2)^2 - (\sigma_j^2)^2
\]

(6) means the constraint that the performance gains can be achieved by using co-location information fusion. Only if the condition (6) is satisfied, two users can be called as “co-location”. If the initial measurement variance of user \( i \) and \( j \) are approximately the same, i.e., \( \sigma_i^2 = \sigma_j^2 = \sigma \). Then (6) can be simplified as \( d_{ij}^p < \sqrt{2}\sigma \), where \( d_{ij}^p = \| \mathbf{p}_j - \mathbf{p}_i \| \) is the calculated relative distance by using the measured GPS position. Since \( \sigma^2 = (\sigma^2)^2 + (\sigma^2)^2 \), if \( \sigma = \sigma^2 = \sigma^2 \), then \( d_{ij}^p < \sqrt{2}\sigma^2 = 2\sigma \).

The relation between maximum allowable distance and measurement variance is shown in Fig. 2a: the performance gains with regard to the relative distance and variance is shown in Fig. 2b. Note that \( d_{ij}^p \) is different from the ranging measurement \( d_{ij} \); \( d_{ij}^p \) is obtained by fusing GPS positions of smartphones while \( d_{ij} \) is obtained by inter-user ranging. In addition, \( d_{ij} \) is the unknown true distance between Node \( i \) and Node \( j \).

IV. COOPERATIVE LOCALIZATION WITH RELATIVE DISTANCE CONSTRAINT

The cooperative localization scheme in the previous section is restricted by a maximum allowable distance specified by (6), i.e., the scheme is ineffective for the cases that violate (6). To further improve the positioning accuracy, in this section, we develop a cooperative localization scheme that leverages relative distances among the smartphones.
A. Mathematical Modeling

For a pair of collaborators $\mathbf{p}_i$ and $\mathbf{p}_j$, their Euclidean distance can be denoted as $d_{ij} = ||\mathbf{p}_i - \mathbf{p}_j|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where $|| \cdot ||$ is the 2-norm of the vector. Considering the measurement error, the estimated distance between collaborators is the noise version of $d_{ij}$ as $\hat{d}_{ij} = d_{ij} + n_{ij}$, where $n_{ij}$ is a Gaussian noise component with $n_{ij} \sim \mathcal{N}(0, \sigma^2_{ij})$. The term of $\hat{d}_{ij}$ is a range bias induced by non-line-of-sight (NLOS) propagation, and $b_{ij} = 0$ when the measurement is in line-of-sight (LOS) condition. In real situations, the inter-node distance information is not fully available, i.e., some of the measurements are missing or unavailable. To deal with such condition, we define distance measurement matrix as $\mathbf{D} = \{d_{ij} : (i, j) \in \mathcal{N}_g\}$ with $d_{ij} = 0$ represents the unavailable measurements. The matrix $\mathbf{D}$ is a sparse matrix with sparse rate $\gamma$ defined as the number of $d_{ij} = 0$ terms divided by the total number of $m(m - 1)/2$.

Fisher information $J$ (the reciprocal of CRLB) is often used as a metric to assess the accuracy of a particular position estimation. Hence, parameters to be estimated are the collaborator’s refined position $\hat{\mathbf{p}}_k = (\hat{x}_k, \hat{y}_k)^T$, $k \in \mathcal{N}_g$ by using their initial position and relative distance. For notational convenience, we denote the unknown parameter as $\theta = [\hat{\mathbf{p}}_k]$, where $1 \leq k \leq N_g$. Let $\hat{\theta}$ denotes an estimation of the parameter $\theta$. The error covariance matrix of $\hat{\theta}$ satisfies Information Inequality as

$$\mathbb{E}_r \{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \} \geq J_\theta^{-1}$$

where $J_\theta$ is the Fisher information matrix (FIM) of non-random parameter $\theta$.

The joint likelihood ratio of the discrete random vector $r$ of the received signal and random parameter $\theta$ can be shown as

$$f(r, \theta) = f(r|\theta) \cdot g(\theta),$$

where $f(r|\theta)$ is the conditional pdf, $g(\theta)$ is the a priori probability density function of $\theta$. The generalized Fisher Information Matrix (FIM) for $\theta$ is given by

$$J_\theta = \mathbb{E}_r \left\{ \frac{\partial}{\partial \theta} \ln(f(r, \theta)) \cdot \left( \frac{\partial}{\partial \theta} \ln(f(r, \theta)) \right)^T \right\}$$

(8) can be further decomposed,

$$J_\theta = J_{f(r, \theta)_{j\neq i}} + J_{f(r, \theta)_{j=i}} + J_{g(\theta)}$$

where the first term indicates the position information from a collaborator using GPS; the second term indicates the inter-ranging information between collaborator $i$ and $j$; and the third term denotes a priori information on $\theta$.

From (9), we know that the cooperative localization contributes to the second term; the resulting FIM can be much better than conventional localization methods that just use a priori information and $j = i$ term. By using the initial GPS position result and inter-node information as prerequisite, perform post-decision optimization can obtain a more accurate position result $\hat{\mathbf{p}}_k = (\hat{x}_k, \hat{y}_k)^T$, $k \in \mathcal{N}_g$.

B. Sparse Steepest Descent Optimization

The calculated GPS position for user $i$ is $\hat{\mathbf{p}}_i$ with measurement noise variance as $\sigma^2_{ij}$. The distance between user $i$ and $j$ can be calculated by $d_{ij} = ||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j||$. The measured distance between user $i$ and $j$ is $\hat{d}_{ij}$ with the true value $d_{ij}$ and the variance $\sigma^2_{ij}$.

With two independent measurements $\hat{\mathbf{p}}_i$ and $\hat{d}_{ij}$ available, the problem can be described as to refine the position $\hat{\mathbf{p}}_j$ by utilizing the relative ranging information $\hat{d}_{ij}$. Typically, the ranging accuracy of $\hat{d}_{ij}$ is more accurate than the GPS positioning accuracy due to the short distance between users. We use the following neighborhood-based weighted least-squares estimation to improve the positioning accuracy of $\hat{\mathbf{p}}_i$, $\forall i$, i.e., minimizing the squared error between the calculated distance and the measured distance:

$$\hat{\mathbf{p}} := \arg \min_{\hat{\mathbf{p}}} e(\hat{\mathbf{p}}) = \arg \min_{\hat{\mathbf{p}}} \sum_{(i, j) \in \mathcal{N}_g} \mu_{ij}(||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j|| - \hat{d}_{ij})^2$$

(10)

where $e(\hat{\mathbf{p}})$ is the total sum of distance errors between all the users, and $\mu_{ij}$ is a weight that is inversely proportional to the variance $\sigma^2_{ij}$. $\hat{\mathbf{p}}$ is a matrix whose columns are $\hat{\mathbf{p}}_i$, $i \in \mathcal{N}_g$, where $\mathcal{N}_g$ is the set of all the collaborators in a neighborhood.

The objective function of (10) achieves the minimum value when the total distance calculated by GPS position equals to the measured distance, i.e., more accurate results of position is achieved at the level of the ranging accuracy. To solve the optimization problem of (10), we apply steepest descent method to reduce the error function and calculate the updated version of user position.

Perform the gradient operation $\nabla$ of the error function $e(\hat{\mathbf{p}})$ as

$$\nabla_i e(\hat{\mathbf{p}}) = 2 \sum_{(i, j) \in \mathcal{N}_g} \mu_{ij}(||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j|| - \hat{d}_{ij})\nabla_i(||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j|| - \hat{d}_{ij})$$

(11)

where $\hat{d}_{ij}$ is a measurement value, $\nabla_i \hat{d}_{ij} = 0$. $||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j||$ represents the distance from $\hat{\mathbf{p}}_i$ to $\hat{\mathbf{p}}_j$, i.e., $||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. The gradient of such distance can be written as $\nabla_i ||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j|| = (\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)/||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j||$. Then (11) can be calculated as

$$\nabla_i e(\hat{\mathbf{p}}) = 2 \sum_{(i, j) \in \mathcal{N}_g} \mu_{ij}(||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j|| - \hat{d}_{ij}) \nabla_i(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)$$

where $\hat{d}_{ij} = \hat{d}_{ij}/||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j||$ is the normalized relative distance; it also characterizes the difference between measured distance and calculated distance from position. After optimization, $\hat{d}_{ij}$ should approaching to 1. $\mathcal{N}_g^+$ represents the sparse set that $\hat{d}_{ij} \neq 0$. The relative ranging results between users are not fully available that some measurements of $\hat{d}_{ij}$ are missing, i.e., $\hat{d}_{ij} = 0$. The sparse property of the distance matrix
\( D = \{d_{ij} : (i, j) \in \mathcal{N}_g\} \) causes the performance gains contributed by distance restraint not fully available especially when the sparse rate \( \gamma \) is high. However, such sparse feature can be utilized to speedup the processing by using sparse matrix operation.

After obtaining the gradient function of the error function \( e(\hat{P}) \), the new position can be updated by using

\[
\hat{P} := \hat{P} + \alpha \nabla_i e(\hat{P})
\]

where \( \alpha \) is the iterative step size and \( \alpha \in (0, 1] \). Eq. (13) should be imported column-wisely as \( \hat{P}_i := \hat{P}_i + \alpha \nabla_i e(\hat{P}) \), \( \forall i \) with \( \hat{P}_i = (\hat{x}_i, \hat{y}_i)^T \).

The steepest descent approach is a local optimization method with strong requirement of the initial value selection. However, for our application that GPS position results can be used as the initial value, the overall performance of steepest descent can be guaranteed to provide an optimized value of the position under the restraint of the relative distance measurement.

### C. Weighting Center based Polar Optimization

In the previous subsection, the optimized position results are achieved by minimizing the error between \( ||\hat{p}_i - \hat{p}_j|| \) and measured distance \( d_{ij} \). The optimization process is utilizing the gradient iteration. Another feasible approach is assume the measured distance accurate and replace the true relative distance with \( d_{ij} \). The weighting center between two users' position is more accurate than the individual results. Then update \( \hat{p}_i := f(\hat{p}_i, d_{ij}) \) with the relative distance and weighting center.

The relation to the position of user \( i \) and \( j \) can be expressed as \( d = ||p_i - p_j|| \). For the measured relative distance \( \hat{d}_{ij} \), \( d \) can be replaced by \( \hat{d} \triangleq d_{ij} \). The initial position measurement \( \hat{p}_i \) follows Gaussian distribution with mean value of \( p_i \). The weighting center of position \( \hat{p}_i \) and \( \hat{p}_j \) is theoretically more stable because random deviation can be canceled out with high probability. Denote the weighting center \( \hat{p}_{ij}^w = (\hat{x}_{ij}^w, \hat{y}_{ij}^w)^T \), \( \hat{p}_i = (\hat{x}_i, \hat{y}_i)^T \), \( \hat{p}_j = (\hat{x}_j, \hat{y}_j)^T \). The angle from the position of node \( i \) to node \( j \) is estimated as

\[
\hat{\theta} = \arctan(y_j - y_i)/(x_i - x_j)
\]

With the weighting center and \( \theta \) available, the node position \( i \) and \( j \) can be re-estimated in the Polar-coordinate domain. The position of user \( j \) can be calculated by transferring the Polar-coordinate to Cartesian coordinate by

\[
\begin{align*}
\hat{x}_i &:= \hat{x}_{ij}^w + a_x d/2 \cdot \cos(\hat{\theta}) \\
\hat{y}_i &:= \hat{y}_{ij}^w + a_y d/2 \cdot \sin(\hat{\theta}) \\
\hat{x}_j &:= \hat{x}_{ij}^w - a_x d/2 \cdot \cos(\hat{\theta}) \\
\hat{y}_j &:= \hat{y}_{ij}^w - a_y d/2 \cdot \sin(\hat{\theta})
\end{align*}
\]

where \( a_x \) and \( a_y \) are the unit vector from the direction of node \( i \) to \( j \), with equation as \( a_x = (x_i - x_j)/|x_i - x_j| \) and \( a_y = (y_i - y_j)/|y_i - y_j| \).

For every iteration process, we need to use the measured position results of \( \hat{p}_i \) and \( \hat{p}_j \) to update the weighting centering \( \hat{P}_w \) and \( \hat{\theta} \). The coefficient of updating is chosen as \( (W_m + n - 1)/(W_m + n) \), where \( W_m \) is the window length, \( n \) is the iteration step. Then, substitute \( \hat{p}_{ij}^w \) and \( \hat{\theta} \) in (15) with new estimated, the optimized position results for node \( i \) and \( j \) are obtained.

Different from the calculation of (11) that performs over all the available nodes of \( \sum_{(i,j)\in\mathcal{N}_g}^n \), (15) only process for two users, i.e., user \( i \) and \( j \). Through perform such pair-wise optimization over the whole sparse set \( \mathcal{N}_g \), the positions for all the users can be optimized.

### V. Numerical Results

#### A. Cooperative Localization Under Co-location

To illustrate the clustering result of the co-location users, and evaluate the performance of position optimization for clusters, we conduct monte-carlo simulation to calculate the error CDF by changing the noise variance of initial position results.

The \( (x, y) \) coordinates of the positions of nine users (smartphones) are shown in Fig. 3a; the positions of each user follow the same two-dimensional Gaussian distribution and are shown by the same color in Fig. 3a. As shown in Fig. 3a, there are three clusters, each of which consists of three users; each user has 200 positions, which may be due to measurement error or slight movement of the user; so the total number of positions is 1800. Using the protocol in Section III-B, we can obtain an affinity matrix of dimension 1800 by 1800. We apply the
normalized cuts algorithm [15] algorithm to the affinity matrix and obtain the clustering results shown in Fig. 3b. Fig. 3b shows the affinity matrix; the darker of the entry at Row $i$ and Column $j$, the larger similarity measure between Position $i$ and Position $j$. The affinity matrix shown in Fig. 3b is permuted according to the clustering results, i.e., Position 1 to Position 600 belong to Cluster 1, Position 601 to Position 1200 belong to Cluster 2, and Position 1201 to Position 1800 belong to Cluster 3. As shown in Fig. 3b, all the positions are correctly grouped into the correct cluster and the nine users are correctly grouped into three clusters, i.e., the positions with large similarity measures are grouped together.

The users in the same clusters can perform location fusion to improve their individual location accuracy can be improved by cooperating with other cluster members. The CDF result is shown in Fig. 3c with initial position variance of $\sigma = 0.2$ and $\sigma = 0.6$. The curve of “Init” is the CDF of initial position accuracy; the “Col” is the CDF of position accuracy after cooperation. From Fig. 3c, we know that the performance improvement of social collaboration is significant.

### B. Cooperative Localization With Relative Distance Constraint

1) Sparse Steepest Descent Optimization: To evaluate the performance of our proposed sparse steepest descent (SSD) optimization, we conduct monte-carlo simulation of 12 users with circular initial positions. The initial measurement is very noisy as shown in Fig. 4a. When the inter-user ranging information is available, the position optimization results is shown in Fig. 4b and Fig. 4c with sparse rates of $\gamma = 0.73$ and $\gamma = 0$, respectively. The position accuracy has been greatly improved even with very sparse ranging information.

The mean CDF curves for 12 users of various approaches and different sparse rates are shown in Fig. 5 with initial measurement variance of $\sigma = 0.3$, real position radius $R = 0.5$. The “MA” represents the conventional moving average method used for the initial measurements. Even when the ranging sparse rate is very high, i.e., $\gamma = 0.73$, the performance superiority over “MA” is still sufficient. Another interesting point lies in the no performance degradation when sparse rate is lower than $\gamma = 0.4$. Such property can help reduce the overall ranging costs while maintaining desired performance gains.

### C. Jointly Optimized Approach

The performance of the cooperative localization using ranging information can be even improved when combine our proposed Sparse Steepest Descent Optimization and Weighting Center based Polar Optimization together. Since Polar based optimization is performed for two users, i.e., in a local way, we execute the Polar method after the the global SSD approach. The measurement results are shown in Fig. 6. “Initial” is the initial position measurement; “SSD” case is using our proposed Sparse Steepest Descent Optimization; “SSD+Polar” is using the Polar optimization after the SSD processing.

The CDF figure is shown in Fig. 8. We can know that using Polar and SSD optimization, the performance gains are larger than using the conventional moving average method. When combine SSD and Polar together, the performance can be even improved as shown in Fig. 8.

### VI. EXPERIMENTAL VALIDATION

#### A. Experiment Setup

We conducted experiments by using Android phones to collect location data, and validate our proposed cooperative localization technique by using these real measured results. The data is collected by using Google Nexus S smartphones.
To facilitate the data processing and avoid using large-scale geo-coordinates, we convert the longitude and latitude value to the cartesian coordinate (x,y,z) under the standard of Geodetic Reference System 1980 (GRS80). By subtracting an offset from the obtained (x,y) data, a regional data set can be obtained.

B. Social Cooperative Position Optimization

To evaluate the performance of our proposed cooperative location optimization approach for multi-users in real environments, we conduct measurements for 9 users with random positions in campus environments. The initial measurement results are shown in Fig. 7a. From Fig. 7a, we know that the initial GPS localization results are very noisy due to the blockage and interference of the satellite signal. Different from the simulation results, the obtained GPS results show strong coherence among adjacent measurements. To demonstrate the performance gains contributed by our schemes, we perform social-aided cooperative processing under the co-location and relative distance constraint. For convenience, we denote “Init” as the initial position results; “Col” is the result obtained by utilizing the co-location information. “Polar”, “SSD” and “Polar+SSD” are our proposed schemes by using the ranging-based information for collaboration. We follow the same terms/notations used in Section V.
We apply the normalized cuts algorithm [15] algorithm to the affinity matrix corresponding to Fig. 7a, and obtain the clustering results shown in Fig. 9. The process is the same as that in Section V-A. As shown in Fig. 9, all the positions are correctly grouped into the correct cluster and the nine users are correctly grouped into four clusters, i.e., the positions with large similarity measures are grouped together.

The CDF results of using different algorithms are shown in Fig. 10. We observe that the conventional moving average “MA” approach does not show performance improvement over the initial position results due to the dependency between adjacent measurements. By clustering the 9 users into 4 clusters with co-location, and perform location fusion, the location accuracy of “Col” is much better than “MA” as shown in Fig. 10.

If the relative distance information can be obtained, the accuracy can be even improved by using “Polar” and “SSD” approaches. The scatter figure of using “SSD” is shown in Fig. 7b. After perform joint optimization of “SSD+Polar”, the more accurate results are shown in Fig. 7c. By comparing the results of Fig. 7c to the initial measurement results the performance improvement of using “SSD+Polar” is significant.

From the statistical results of Fig. 10, and using 80% probability as an example, the initial GPS accuracy is around 5 m. After using co-location based cooperative processing, the achieved accuracy is about 4 m. When using our proposed ranging-based optimization approach “SSD+Polar”, the positioning accuracy is approximately 1.2 m. These results demonstrate the effectiveness of our proposed social cooperative localization scheme.

VII. CONCLUSIONS

The GPS receiver of a smartphone does not produce accurate position and does not work in harsh environments such as indoor environment. In addition, the GPS receiver onboard is inefficient in power consumption. To address positioning inaccuracy and power inefficiency, in this paper, we proposed two social-aided cooperative localization schemes. Specifically, the first scheme fuses the GPS positions of multiple co-located smartphones in a social network when relative distances between smartphones are not available; and the second scheme uses neighborhood-based weighted least-squares estimation when relative distances between smartphones are available. The energy efficiency is achieved by sharing location information among co-located users so that only a small set of smartphones need to turn on GPS receivers at any time. Numerical and experimental results conclusively demonstrate that our proposed cooperative localization schemes can achieve considerable performance gain in both indoor and outdoor environments. For example, in the experiments of 9 users with random positions, when relative distances are available, the positioning accuracy of our scheme is 1.2 m with a confidence level of 80%. In contrast, a regular GPS receiver has an accuracy of 4.7 meters with a confidence level of 80%. Our future work will further enhance the accuracy of cooperative localization and make our smartphone app available for more location-based or location-aware services and applications.

REFERENCES