

# Deploying Sensors for Gravity Measurement in a Body-Area Inertial Sensor Network

Chun-Hao Wu and Yu-Chee Tseng

**Abstract**—This paper deals with human posture tracking by deploying accelerometers on a human body. One fundamental issue in such scenarios is how to calculate the gravity. This is very challenging especially when the human body parts keep on moving. Fortunately, it is likely that there is a point of the body that touches the ground in most cases. This allows sensors to collaboratively calculate the gravity vector. Assuming multiple accelerometers being deployed on a rigid part of a human body, a recent work proposes a data fusion method to estimate the gravity vector on that rigid part. However, finding the optimal deployment of sensors that minimizes the estimation error of the gravity vector is not addressed. In this paper, we formulate the deployment optimization problem and propose two heuristics, called *Metropolis-based method* and *largest-inter-distance-based method*. Simulation and real experimental results show that our schemes are quite effective in finding near-optimal solutions for a variety of rigid body geometries.

**Index Terms**—Accelerometer, deployment optimization, modeling of systems and physical environments, wireless sensor network.

## I. INTRODUCTION

A BODY-AREA inertial sensor network (BISN) consists of multiple accelerometers, magnetometers, and gyroscopes connected by wired/wireless links. An important usage of a BISN is to track human motions through these inertial sensors. Its applications include video games [1], robotic balancing [2], localization [3], sports training [4], medical care [5], [6], and computer graphics [7].

Regarding a human body as multiple rigid parts connected by joints, many researchers consider a full-body motion as the combination of the motions of multiple rigid parts through the human body skeleton structure. Basically, the motion of each rigid part is measured by some inertial sensors deployed on a human body [8]. Many gesture applications have been developed [5], [9], [10].

Manuscript received July 9, 2012; revised November 22, 2012; accepted December 9, 2012. Date of publication December 21, 2012; date of current version March 26, 2013. The work of Y.-C. Tseng was supported in part by MoE ATU Plan under NSC Grant 98-2219-E-009-019, Grant 98-2219-E-009-005, and Grant 99-2218-E-009-005, the Academia Sinica under Grant AS-102-TP-A06, the ITRI of Taiwan, the D-Link, and the MoEA of China through the AMI Enhancement Project of III. The associate editor coordinating the review of this paper and approving it for publication was Prof. Roozbeh Jafari.

C.-H. Wu is with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan.

Y.-C. Tseng is with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan, and also with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSEN.2012.2235143

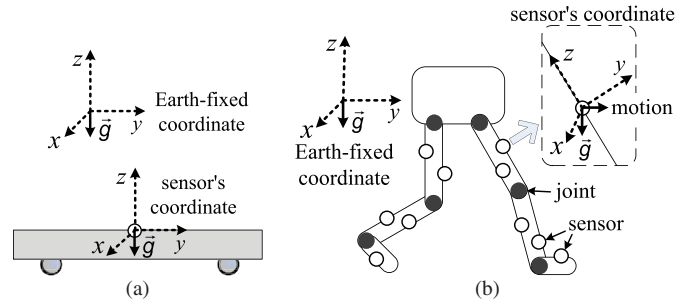


Fig. 1. Gravity measurement in (a) car and (b) human body, where  $g$  is the gravity vector.

One fundamental issue in BISNs is the *gravity measurement problem* [8]. Accelerometers sense both motion acceleration and the gravity simultaneously. It is desirable to obtain only the gravity vector with respect to a rigid part of a human body, no matter it is moving or not. A trivial case is when an accelerometer is mounted on a car driving on a flat plane as in Fig. 1(a). The gravity vector  $\vec{g}$  with respect to the accelerometer is always the same. However, as shown in Fig. 1(b), for a human body, the orientation of each rigid part keeps on changing. Therefore, for accelerometers, determining the gravity vector of a rigid part is a fundamental issue to be solved before trying to track human motions.

To solve the gravity measurement problem, [2] considers a rigid part of a human body as being held by a joint which is at a constant velocity with respect to the Earth coordinate. Assuming multiple accelerometers deployed at known locations on the rigid part, it proposes a data fusion method to calculate the gravity vector on the rigid part. However, it only partly addresses the error issue, which is common to sensor reading. Also, the locations of these sensors may highly impact the potential value of errors. In this paper, based on the model in [2], we further formulate the *deployment optimization problem* as one of finding the best locations of accelerometers on a rigid part that minimize the estimation error of the gravity vector. The problem is difficult even for simple geometries of rigid parts. We propose two heuristics, called *Metropolis-based method* and *largest-inter-distance-based (LID-based) method*. The former follows a probabilistic search principle that is theoretically guaranteed to reach near-optimal solutions eventually. The latter is based on a “largest inter-distance (LID)” guideline obtained from our observations, which states that sensors should be evenly distributed over the surface of the rigid part. We show that this guideline leads to an optimal

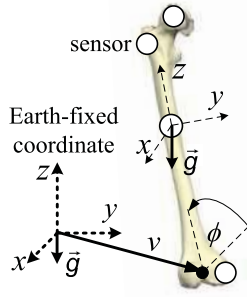


Fig. 2. Geometrical model of our work, where  $v$  and  $\phi$  are unknown to our system and  $g$  is to be calculated.

solution for some special cases and near-optimal solutions in general. Experimental results show that both methods perform quite well, even for complicated geometries.

We remark that although the LID guideline, which tries to push accelerometers away from each other on one rigid part, has been widely adopted in other fields, its validity on our sensing model is not yet verified. Our analysis starts from the kinematical model of rigid parts and we demonstrate its effectiveness simulation- and experiment-wise.

Our results are applicable to any geometrical model illustrated in Fig. 2, where a rigid part is held by a joint that moves at a constant velocity  $v$  with respect to an Earth-fixed coordinate. The rigid part may rotate at any angular velocity  $\phi$  about the joint. Both  $v$  and  $\phi$  are unknown to our system. This allows us to apply it to many physical rehabilitation, robotic balancing, and gait analysis scenarios. Also note that in this paper, we do not address physical deployment issues, such as the fixation of accelerometers on a human body. They are addressed by many emerging techniques and are beyond our scope [11].

The rest of this paper is organized as follows. Section II reviews related works. Section III formulates the gravity measurement and the deployment optimization problems. Section IV presents our heuristics. Performance studies are discussed in Section V. Conclusions are drawn in Section VI.

## II. RELATED WORKS

The deployment of sensors directly determines the network topology and influences communications. This issue has long been studied in wireless sensor networks. A comprehensive survey can be found in [12]. Many deployment algorithms have been proposed for isotropic sensors with unit-disc sensing coverage [13]–[16]. Directional sensors, such as cameras, have been studied in [17] and [18]. The irregularity of sensing quality has been studied in [19]. However, none of these studies is suitable for accelerometers when they are deployed on a human body for motion sensing.

Comprehensive surveys on BISNs can be found in [20] and [21], and surveys on human motion tracking can be found in [22] and [23]. Traditionally, one accelerometer is deployed on one rigid part of a human body to measure its orientation, and the results from multiple parts are combined through a skeleton model to reconstruct full-body motions [3], [8], [24]. With additional motion constraints,

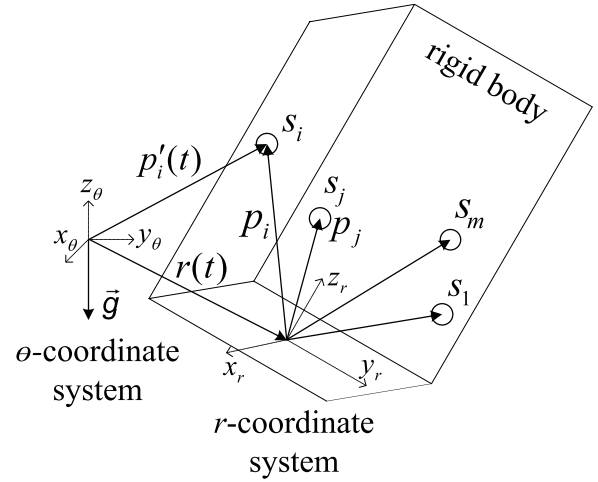


Fig. 3. Kinematical model for one rigid body.

[25]–[28] further avoid infeasible orientations. However, these approaches rely on empirical methods to eliminate such factors. To solve this issue, [2] assumes multiple accelerometers being deployed on one rigid part and proposes a kinematical method that optimally measures the gravity. However, how to optimize the locations of accelerometers so as to remove as much sensor reading errors as possible is left by as an open problem.

## III. GRAVITY MEASUREMENT AND DEPLOYMENT OPTIMIZATION PROBLEMS

This paper deals with human posture tracking by deploying accelerometers on a human body. One fundamental issue in such scenarios is how to calculate the gravity, no matter when the body parts are moving or not [8]. A human body can be regarded as multiple movable *parts*, each being a *rigid body*, connected to another part by a *rotational joint* [29], [30]. Multiple accelerometers are placed on each rigid body for gravity measurement.

To formulate the *gravity measurement problem*, we consider one rigid body deployed with multiple sensors as in Fig. 3. We assume that there is an Earth-fixed coordinate system, representing views of a fixed observer, with  $\theta = (0, 0, 0)$  as its origin and  $x_\theta$ ,  $y_\theta$ , and  $z_\theta$  as its axes. The gravity  $\vec{g}$  with respect to this  $\theta$ -coordinate is thus  $1\text{ g}$  (or  $9.8\text{ m/sec}^2$ ) along the  $-z_\theta$  axis. Let the joint of the rigid body be at location  $r(t)$  with respect to the  $\theta$ -coordinate at time  $t$ . For the rigid body, we assume that there is a fixed coordinate with respect to the joint with  $r(t)$  as its origin and  $x_r$ ,  $y_r$ , and  $z_r$  as its axes. Therefore, each point on the rigid body has a fixed coordinate with respect to the  $r$ -coordinate, no matter the rigid body moves or not. For any point at location  $p$  with respect to the  $r$ -coordinate, its location with respect to the  $\theta$ -coordinate at time  $t$  can be written as  $p'(t) = r(t) + R(t)p$ , where  $R(t)$  is the  $3 \times 3$  *rotation matrix* that rotates  $(x_r, y_r, z_r)$  to  $(x_\theta, y_\theta, z_\theta)$  [29].

Let  $s_1, s_2, \dots, s_m$  be  $m$  accelerometers deployed on the rigid body and  $p_1, p_2, \dots, p_m$  be their locations at the  $r$ -coordinate, respectively. Without loss of generality, we assume that these

accelerometers are properly pre-calibrated in the sense that their  $x$ -,  $y$ -, and  $z$ -axes are aligned to the  $x_r$ ,  $y_r$ , and  $z_r$  axes of the  $r$ -coordinate, respectively (otherwise, a rotation matrix from the  $r$ -coordinate to each sensor's coordinate would do the translation). This implies that the gravity being observed by the joint is the same as that being observed by any sensor.

Now, consider any sensor  $s_i$ ,  $i = 1, 2, \dots, m$ . Its location with respect to the  $\theta$ -coordinate at time  $t$  is  $p'_i(t) = r(t) + R(t)p_i$  (note that  $p_i$  is time-invariant). Taking the second derivative of  $p'_i(t)$ , we have its acceleration with respect to the  $\theta$ -coordinate

$$\ddot{p}'_i(t) = \ddot{r}(t) + \ddot{R}(t)p_i. \quad (1)$$

Since  $s_i$  is calibrated to the  $r$ -coordinate, its actual reading  $a_i(t)$  should be  $a_i(t) = R^T(t)(-\ddot{p}'_i(t) + \vec{g}) + n_i(t)$ , where  $R^T(t)$  is the inverse/transpose of  $R(t)$  (the inverse of a rotation matrix equals its transpose) and  $n_i(t)$  is the noise. Note that both  $a_i(t)$  and  $n_i(t)$  are  $3 \times 1$  vectors with respect to the  $r$ -coordinate, and each element of  $n_i(t)$  has a zero mean with a standard deviation  $\sigma_n$ . [For example, when the rigid body is at rest,  $\ddot{p}'_i(t) = 0$  and  $a_i(t) = R^T(t)g + n_i(t)$ ; when it falls freely,  $a_i(t) = n_i(t)$ .]

Below, assuming a fixed  $t$ , we will omit time information in our formulation. Plugging Eq. (1) into  $a_i$ , we have

$$\begin{aligned} a_i &= R^T(-\ddot{r} - \ddot{R}p_i + \vec{g}) + n_i \\ &= [R^T(-\ddot{r} + \vec{g}), -R^T\ddot{R}] \begin{bmatrix} 1 \\ p_i \end{bmatrix} + n_i. \end{aligned}$$

Putting  $m$  equations for all  $s_i$ 's together, we have the equality

$$A = QP + N \quad (2)$$

where

$$\begin{aligned} A &= [a_1 \dots a_m] \in \mathfrak{R}^{3 \times m}, \\ Q &= [R^T(-\ddot{r} + \vec{g}), -R^T\ddot{R}] \in \mathfrak{R}^{3 \times 4}, \\ P &= \begin{bmatrix} 1 & \dots & 1 \\ p_1 & \dots & p_m \end{bmatrix} \in \mathfrak{R}^{4 \times m}, \\ N &= [n_1 \dots n_m] \in \mathfrak{R}^{3 \times m}. \end{aligned}$$

Here,  $A$  and  $P$  are known and  $Q$  is to be determined.  $P$  is called the *deployment matrix* of sensors  $s_1, s_2, \dots, s_m$ .

Assuming that the joint moves at a constant velocity with respect to  $\theta$ , we can plug  $\ddot{r} = 0$  in Eq. (2). Hence,  $Q$ 's first column vector  $R^T(-\ddot{r} + \vec{g}) = R^Tg$ . By estimating  $Q$ , we can determine  $R^Tg$ . Let  $\hat{Q}$  be an estimation of  $Q$ . Following [2], a  $\hat{Q}$  that makes  $\hat{Q} - Q$  as small as possible can be found by  $\hat{Q} = AP^+$ , where  $P^+$  is the Moore–Penrose pseudoinverse of  $P$  when  $m > 4$  and  $P^+ = P^{-1}$  (the inverse of  $P$ ) when  $m = 4$  [31]. This implies that to find  $\hat{Q}$  we need at least four sensors. Since  $\hat{Q} - Q$  is zero-mean, to measure how  $\hat{Q}$  is close to  $Q$ , [2] defines an *error variance*, which solely depends on the deployment matrix  $P$ :

$$\sigma_e^2(P) = 3\sigma_n^2 \sum_{k=1}^4 \frac{1}{\rho_k^2(P)} \quad (3)$$

where  $\rho_k(P)$  is the  $k$ th largest singular value of  $P$ . It is claimed that a smaller  $\sigma_e^2(P)$  implies a more accurate  $\hat{Q}$ . Therefore,

we formulate the *deployment optimization problem* as follows: given a rigid body and  $m$  accelerometers to be deployed on the surface of the rigid body, the goal is to find a deployment matrix  $P$  such that the error variance  $\sigma_e^2(P)$  is minimized. Note that since the  $3\sigma_n^2$  in Eq. (3) is a constant, we only need to focus on the summation part.

#### IV. OPTIMIZATION HEURISTICS

The above formulation has related the sensor deployment problem to one of finding a  $P$  that minimizes  $\sigma_e^2(P)$ . The problem is difficult even for simple geometries. Below, we present two heuristics, called *Metropolis-based* and *largest-inter-distance-based (LID-based)* methods. The former is based on probabilistic optimization, and the latter is based on some observations and guidelines.

##### A. Metropolis-Based Method

We present a Metropolis-based deployment method that can be applied to rigid bodies of arbitrary shapes. It follows a probabilistic search to get rid of local optimal solutions. It, however, has longer search time. This is acceptable for our sensor deployment problem since once a good solution is found, it can be used repeatedly.

In this method, we partition the surface of the rigid body into mesh-like grid points. We say that two deployment matrices  $P_i$  and  $P_j$  are *neighbors* if they differ by exactly one sensor's location and this sensor's locations in  $P_i$  and  $P_j$  are neighboring grids. We define the eight nearest grid points of a grid point as the latter's neighbors. Therefore, each  $P_i$  has up to  $8m$  neighbors. We denote by  $B(P_i)$  the set of  $P_i$ 's neighbors. The algorithm works as follows.

- 1) Select any arbitrary deployment matrix  $P_i$ .
- 2) From  $P_i$ , we choose one of its neighbors, say  $P_j$ , as the next deployment with probability  $q_{ij}$  (discussed below). Note that  $\sum_{j \in B(P_i)} q_{ij} = 1$ .
- 3) Repeat step 2 for a predefined number of times, and output the best deployment (with the smallest  $\sigma_e^2(P)$ ) along the above search path.

We design the transition probability  $q_{ij}$  according to the Metropolis' theorem [32]. The resulting  $q_{ij}$ 's should ensure: if we run the above search process long enough, each deployment  $P$  should be visited by a mathematically stationary distribution  $\pi(P)$  such that

$$\pi(P) = \frac{(1/\sigma_e^2(P))^\lambda}{C} \quad (4)$$

where the parameter  $\lambda > 3$  and  $C = \sum_P (1/\sigma_e^2(P))^\lambda$ . Note that  $C$  normalizes the distribution, and a higher  $\lambda$  ensures that a better deployment will be allocated a higher visiting probability. By Metropolis' theorem, this transition probability should be defined as

$$q_{ij} = \begin{cases} \frac{1}{8m} \min\{1, \frac{\pi(P_j)}{\pi(P_i)}\} & \text{if } P_j \in B(P_i), \\ 0 & \text{if } P_j \notin B(P_i), \\ 1 - \sum_{P_j \neq P_i} q_{ij} & \text{if } P_j = P_i. \end{cases} \quad (5)$$

Note that the actual value of  $C$  is immaterial in Eq. (5).

We argue that, though simple, this algorithm is highly probable to obtain a near-optimal solution. This is because Eq. (4) is devised similar to a (continuous) Pareto distribution with a density function  $f(x) = C'/x^\lambda$  defined over  $a \leq x$ , where  $C'$  and  $a$  are constants. The Pareto distribution has a mean of  $(\lambda - 1)a/(\lambda - 2)$ , which is very close to the minimal value  $a$  for a large  $\lambda$ . Through this similarity, we can expect the mean  $\sum_P \sigma_e^2(P)\pi(P)$  to be very close to the minimum  $\sigma_e^2$ . Note that the variance of a Pareto distribution is also very tight for a large  $\lambda$ . Hence, once the search reaches the stationary distribution, the output deployment is highly probable to be a near-optimal solution.

### B. Largest-Inter-Distance-Based Method

While the Metropolis-based method takes a long time to search near-optimal solutions, this method is based on some observations and a guideline, making it both efficient and effective in most cases. To minimize the term  $\sum_{k=1}^4 1/\rho_k^2(P)$  in Eq. (3), we propose the following ‘‘largest inter-distance (LID)’’ guideline: *Sensors should be evenly distributed over the surface area such that the distance between each pair of them is maximized.* The guideline is drawn from observations on some special cases where optimal deployments can be found. Below, we discuss these special cases and then present the LID-based method.

To show the optimality of a deployment, we present the following lemma.

*Lemma 1:* The error variance  $\sigma_e^2(P)$  of a  $4 \times m$  deployment matrix  $P = (P_{ij})$  is bounded by

$$\sigma_e^2(P) \geq 16/|P|^2 \quad (6)$$

where  $|P| = \sqrt{\sum_{i,j} P_{ij}^2}$ . The equality holds iff  $\rho_1(P) = \rho_2(P) = \rho_3(P) = \rho_4(P)$ .

*Proof:* The harmonic mean of  $\rho_k^2(P)$ ,  $k = 1, \dots, 4$ , can be expressed by

$$\frac{4}{1/\rho_1^2(P) + 1/\rho_2^2(P) + 1/\rho_3^2(P) + 1/\rho_4^2(P)} = \frac{4}{\sigma_e^2(P)}.$$

By properties of the Frobenius matrix norm [33], we have  $\rho_1^2(P) + \rho_2^2(P) + \rho_3^2(P) + \rho_4^2(P) = |P|^2$ , so the arithmetic mean of  $\rho_k^2(P)$ s is  $|P|^2/4$ . The lemma is a direct consequence of the fact that the arithmetic mean is always larger than the harmonic mean. Eq. (6) holds when  $\rho_i^2(P) = \rho_j^2(P)$  for all  $i$  and  $j$  [34], or simply  $\rho_i(P) = \rho_j(P)$  because  $\rho_i(P) \geq 0$  for all  $i$ . ■

Lemma 1 also suggests that when the joint of a rigid body is at its geometric center, a near-optimal deployment may be obtained according to the LID guideline. To see this, consider a spherical rigid body with its joint at its center. Since each point on the sphere is equidistant to the joint, the  $|P|$  has a fixed value. In Fig. 4(a), we deploy three fixed sensors at  $p_1 = (1/\sqrt{3} - 1, 1 + 1/\sqrt{3}, -\sqrt{3}/3)$ ,  $p_2 = (1 + 1/\sqrt{3}, 1/\sqrt{3} - 1, -\sqrt{3}/3)$ , and  $p_3 = (-2/\sqrt{3}, -2/\sqrt{3}, -\sqrt{3}/3)$ , and move the fourth sensor  $p_4$  around on the upper part of the sphere. Let the coordinate of  $p_4$  be  $(p_{4x}, p_{4y}, p_{4z})$ . Fig. 4(b) shows the reciprocal  $1/\sigma_e^2(P)$  with respect to  $(p_{4x}, p_{4y})$ . Evidently, a larger  $1/\sigma_e^2(P)$ , or a

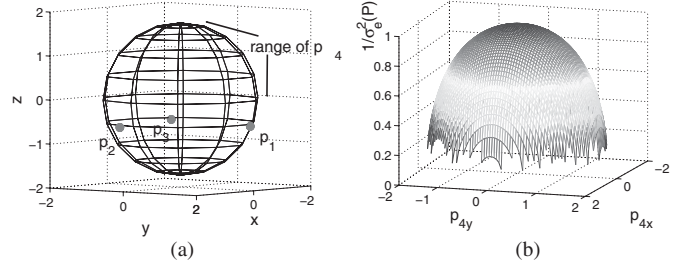


Fig. 4. (a) Sphere rigid body attached by three fixed sensors and one moving sensor and (b) reciprocals  $1/\sigma_e^2(P)$  by moving point  $p_4$ .

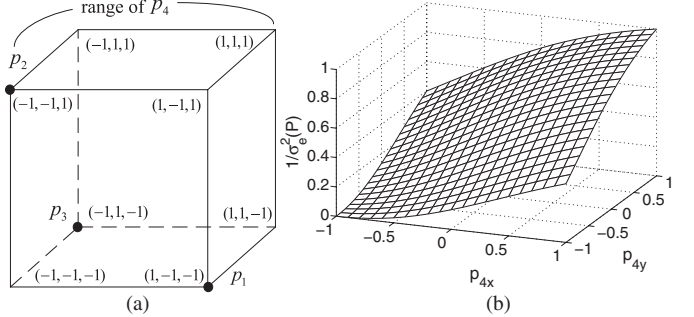


Fig. 5. (a) Cube rigid body attached by three fixed sensors and one moving sensor and (b) reciprocals  $1/\sigma_e^2(P)$  by moving point  $p_4$ .

smaller  $\sigma_e^2(P)$ , can be obtained by moving  $p_4$  farther away from other sensors. The maximal value of  $1/\sigma_e^2(P)$  appears at  $p_4 = (0, 0, \sqrt{3})$ . Under this configuration, we actually have  $\rho_1(P) = \rho_2(P) = \rho_3(P) = \rho_4(P) = 2$ . Since  $|P|$  is fixed for all  $P$ s, the optimality directly follows from Lemma 1. Basically, following the LID guideline to move four sensors around a sphere will result in forming a regular tetrahedron, whose faces are regular triangles. It can be shown that such deployments are optimal (see the proof in Appendix I). We summarize the results with a theorem.

*Theorem 1:* Given four sensors to be deployed on a sphere rigid body whose joint is at its geometric center, a regular tetrahedron deployment gives the minimum  $\sigma_e^2(P)$ .

The LID guideline can also be applied to other geometries. In Fig. 5(a), we consider four sensors to be deployed on a cube whose joint is at its geometric center. We fix three sensors at  $p_1 = (1, -1, -1)$ ,  $p_2 = (-1, -1, 1)$ , and  $p_3 = (-1, 1, -1)$  and move  $p_4 = (p_{4x}, p_{4y}, 1)$  around the top surface where  $-1 \leq p_{4x} \leq 1$  and  $-1 \leq p_{4y} \leq 1$ . Fig. 5(b) shows the reciprocal  $1/\sigma_e^2(P)$  with respect to  $(p_{4x}, p_{4y})$ . As can be seen, when  $p_1, p_2, p_3$ , and  $p_4$  follows the LID guideline and results in forming a regular tetrahedron, the deployment is optimal (see the proof in Appendix II).

*Theorem 2:* Given four sensors to be deployed on a cube rigid body whose joint is at its geometric center, a regular tetrahedron deployment at the vertices gives the minimum  $\sigma_e^2(P)$ .

For the case of a rectangular box with the joint placed at its geometric center, we have experimented many cases with four sensors and found that a tetrahedron-like deployment (such as the one in Fig. 6) always gives the minimum  $\sigma_e^2(P)$ .



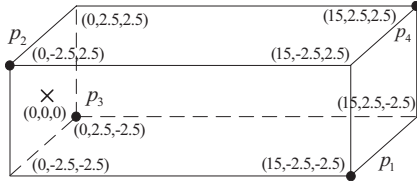


Fig. 6. Rectangular box attached by four sensors with its joint at  $\times$ .

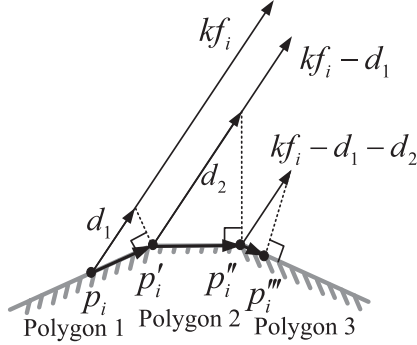


Fig. 7. Example of iterative projection on three consecutive polygons.

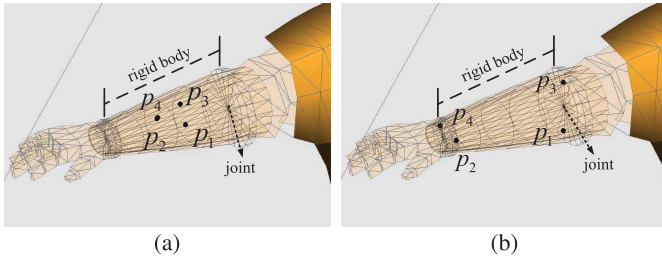


Fig. 8. Example of running the LID-based method by Unity. (a) Initial deployment. (b) Final deployment.

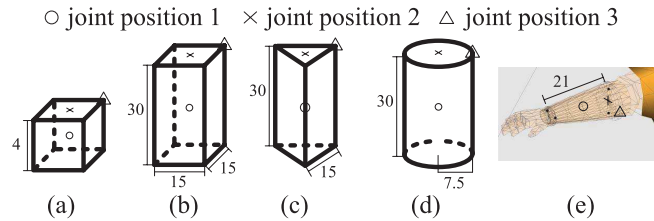


Fig. 9. Rigid body geometries, each with its joint placed at one of the three possible positions. (a) Cube. (b) Box. (c) Prism. (d) Cylinder. (e) Arm.

However, we can not derive a formal proof of its optimality yet. On the other hand, when the joint is not at its geometric center, we do find a counterexample as follows. In the case of Fig. 6, where the joint is at the left surface (marked by a  $\times$ ), the  $\sigma_e^2(P)$  is 0.5844. With the same joint position, when  $p_1 = (0, 2.5, 1.6)$ ,  $p_2 = (0, -0.7, -2.46)$ ,  $p_3 = (1.56, -2.5, 2.48)$ , and  $p_4 = (12.84, 2.5, -2.48)$ , we find a smaller  $\sigma_e^2(P) = 0.5347$ . Hence, the LID guideline is able to give optimal solutions in some special cases and near-optimal solutions in general.

Now, we present an algorithm to implement the LID guideline for rigid bodies of arbitrary shapes. It distributes sensor nodes by introducing virtual forces that mimics the electronic

particles distribution principle: *the nearer the particles, the stronger the repelling forces between them*. This algorithm works as follows.

- 1) Initially, randomly place sensors  $s_1, s_2, \dots, s_m$  on the surface of the rigid body. Let  $p_1, p_2, \dots, p_m$  be their locations, respectively.
- 2) For each pair of  $s_i$  and  $s_j$ , the repelling force contributed by  $s_j$  on  $s_i$  is defined by a  $3 \times 1$  vector

$$f_{ji} = \left( \frac{1}{\|p_i - p_j\|} + \beta \right) \frac{p_i - p_j}{\|p_i - p_j\|}$$

where  $\beta$  is a constant speed to expedite the iterative process. In the exceptional case of  $p_i = p_j$ , we set  $f_{ji}$  to a unit vector of any direction. The total force contributed by all other sensors on  $s_i$  is  $f_i = \sum_{j \neq i} f_{ji}$ .

- 3) For each sensor  $s_i$ , we let it move a displacement vector of  $kf_i$ , where  $k$  is a scaling constant. So the new location of  $s_i$  is  $p_i + kf_i$ . Since this location may not be on the surface of the rigid body, a projection operation is needed (we will discuss this below).
- 4) Let  $p_i^{(old)}$  and  $p_i^{(new)}$  be the previous and next locations of  $s_i$ , respectively. We test whether  $\sum_{i=1}^m \|p_i^{(new)} - p_i^{(old)}\| \leq \sigma_{th}$ , where  $\sigma_{th}$  is a threshold value, or the allowed number of iterations is reached. If so, we terminate this algorithm; otherwise, go back to step 2.

The above algorithm follows the virtual-force discipline [35] to maximize the inter-distances among sensors. Note that since the final locations depend on the initial locations, to avoid falling into a local minimum, we may repeat the algorithm several times with randomized initial locations.

Below, we explain the projection operation in step 3. To be computationally feasible, we approximate a surface by polygons as usually done in computer graphics. Fig. 7 shows an example. Sensor  $s_i$  is at location  $p_i$  and is pushed by a displacement  $kf_i$ . The projection is conducted in a polygon-by-polygon manner. First,  $p_i + kf_i$  is projected to polygon 1. Since the projected point is beyond the range of polygon 1, we regard that a partial displacement  $d_1$  out of  $kf_i$  has been consumed and  $s_i$  has been moved to  $p'_i$ , which is at the edge of polygon 1 (see Fig. 7). A remaining displacement of  $kf_i - d_1$  needs to be applied to  $p'_i$ . Similarly,  $p'_i + (kf_i - d_1)$  is projected to polygon 2. Since the projected point is also beyond the range of polygon 2, we regard that a partial displacement  $d_2$  out of  $kf_i - d_1$  has been consumed and  $s_i$  has been moved to  $p''_i$ , which is at the edge of polygon 2 (see Fig. 7). A remaining displacement of  $kf_i - d_1 - d_2$  needs to be applied to  $p''_i$ . Finally,  $p''_i + (kf_i - d_1 - d_2)$  is projected to polygon 3. Since the projected point is within the range of polygon 3, the final location of  $s_i$  is  $p'''_i$  on polygon 3. Note that the above iterative process is indeed supported by many 3D game engines, such as Unity [36]. Fig. 8 shows how we configure an arm by polygons in Unity and run our LID-based method.

## V. SIMULATION AND EXPERIMENTAL RESULTS

We have conducted simulations and real experiments to verify our schemes.

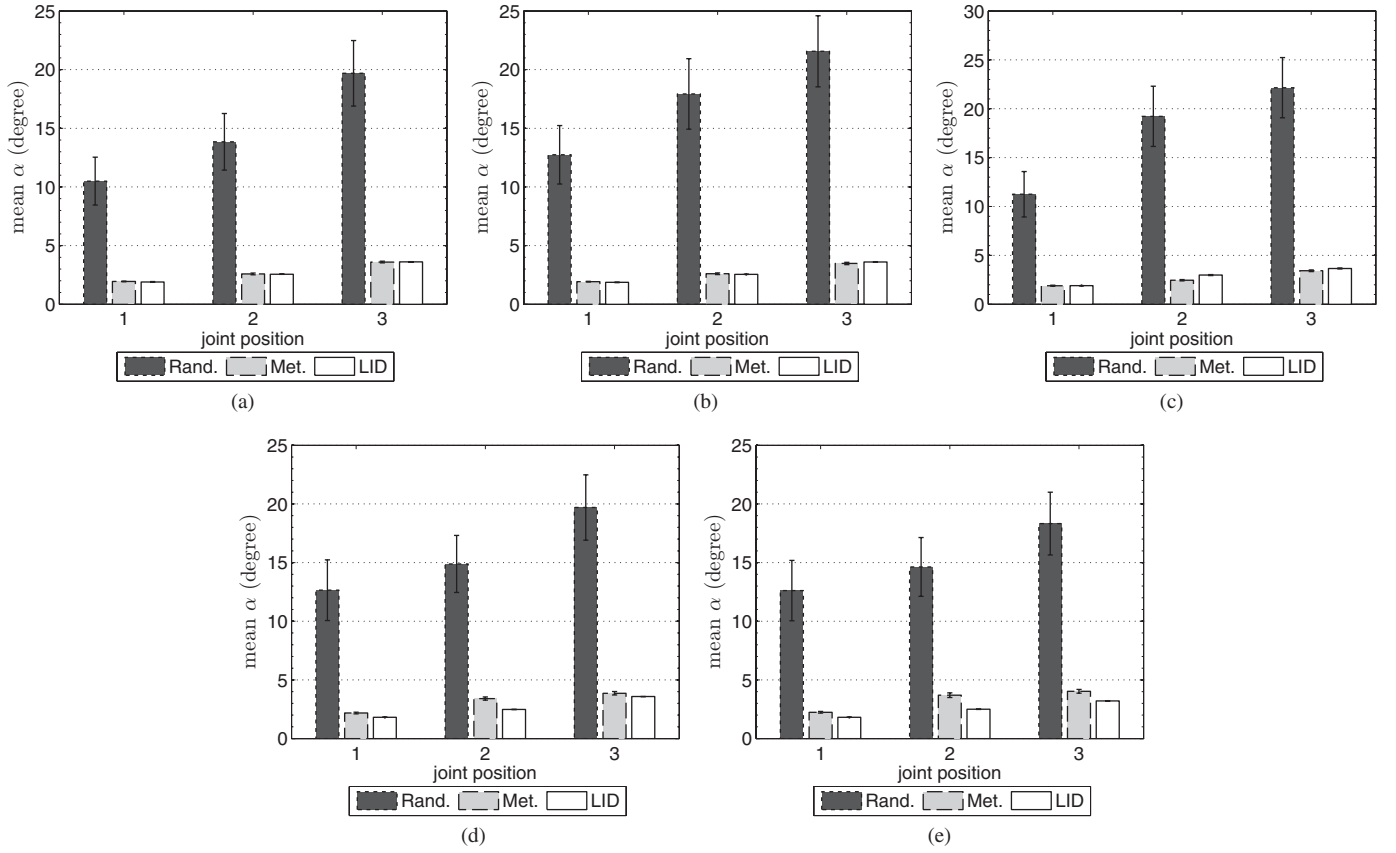


Fig. 10. Comparison of  $\alpha$ , with 95% confidence intervals shown on the bars. (a)–(e) Geometry a–e.

### A. Simulation Setup

We consider five rigid body geometries in Fig. 9. For each geometry, we simulate its joint at three possible positions, one at its geometric center (shown by  $\circ$ ), one at the center of a face (shown by  $\times$ ), and one at a corner (shown by  $\triangle$ ). The combinations of geometries and joints give a total of 15 rigid bodies to be evaluated.

For each rigid body, we compare the following three deployment methods.

- 1) *Random Deployment*: This method distributes sensors at random positions. It represents previous approaches, which place sensors without any optimization [2].
- 2) *Metropolis-Based Deployment*: This is our Metropolis-based deployment method. In our simulations, we give each run 10000 iterations and set the grid sizes of geometries (a) to (e) in Fig. 9 to 0.1, 0.4, 0.4, 0.2, and 0.15, respectively.
- 3) *LID-Based Deployment*: This is our LID-based deployment method. In our simulations, we set  $\beta = 0.5$ ,  $k = 1$ , and  $\sigma_{th} = 0.0001$ . The maximum iteration number of a run is set to 10000.

To simulate the estimation of gravity vectors, for each deployment matrix  $P$ , we follow Eq. (2) to calculate  $\hat{Q}$ . Without any loss of generality,  $R$  in Eq. (2) is set to the identity matrix, so the real gravity vector  $R^T \vec{g}$  (which is hidden in  $Q$ ) becomes  $\vec{g}$ . The  $\vec{R}$  is arbitrarily set to the identity matrix

because it is filtered out by our estimation scheme. The noise level  $\sigma_n$  is set to 50 mg (note that the magnitude of  $\vec{g}$  is 1000 mg). The estimated gravity vector  $\hat{g}$  is the first column of  $\hat{Q}$ . Since  $\vec{g}$  has a constant magnitude, we compare the angle  $\alpha$  between  $\hat{g}$  and  $\vec{g}$ .

### B. Comparison of Accuracy and Execution Time

First, we consider using  $m = 4$  sensors only. For each rigid body, each method is run 200 times (each with its own randomized initial sensor positions) to constitute an averaged simulation result. We compare methods by  $\alpha$ ,  $\sigma_e^2$ , and the execution time.

The results of  $\alpha$  and  $\sigma_e^2$  are shown in Figs. 10 and 11, respectively. The unit of  $\sigma_e^2$  is ignored since it is merely used for comparison. Comparing Figs. 10 and 11, we see that optimizing  $\alpha$  can be done through  $\sigma_e^2$ . The random deployment method always gives large  $\alpha$  and  $\sigma_e^2$ . Our proposed methods obtain significant improvements and also show tight confidence intervals. The LID-based method outperforms the Metropolis-based method in most cases, except a few cases in Fig. 11(c) for the triangular prism. One possible reason for the exceptions may be that it does not consider the joint position. Since both proposed methods give very small  $\sigma_e^2$ , we can believe that their results are near-optimal.

Fig. 12 compares the execution time measured on a PC with an Intel i7 CPU. We report the cases when joints are at the geometric centers. The random method terminates almost

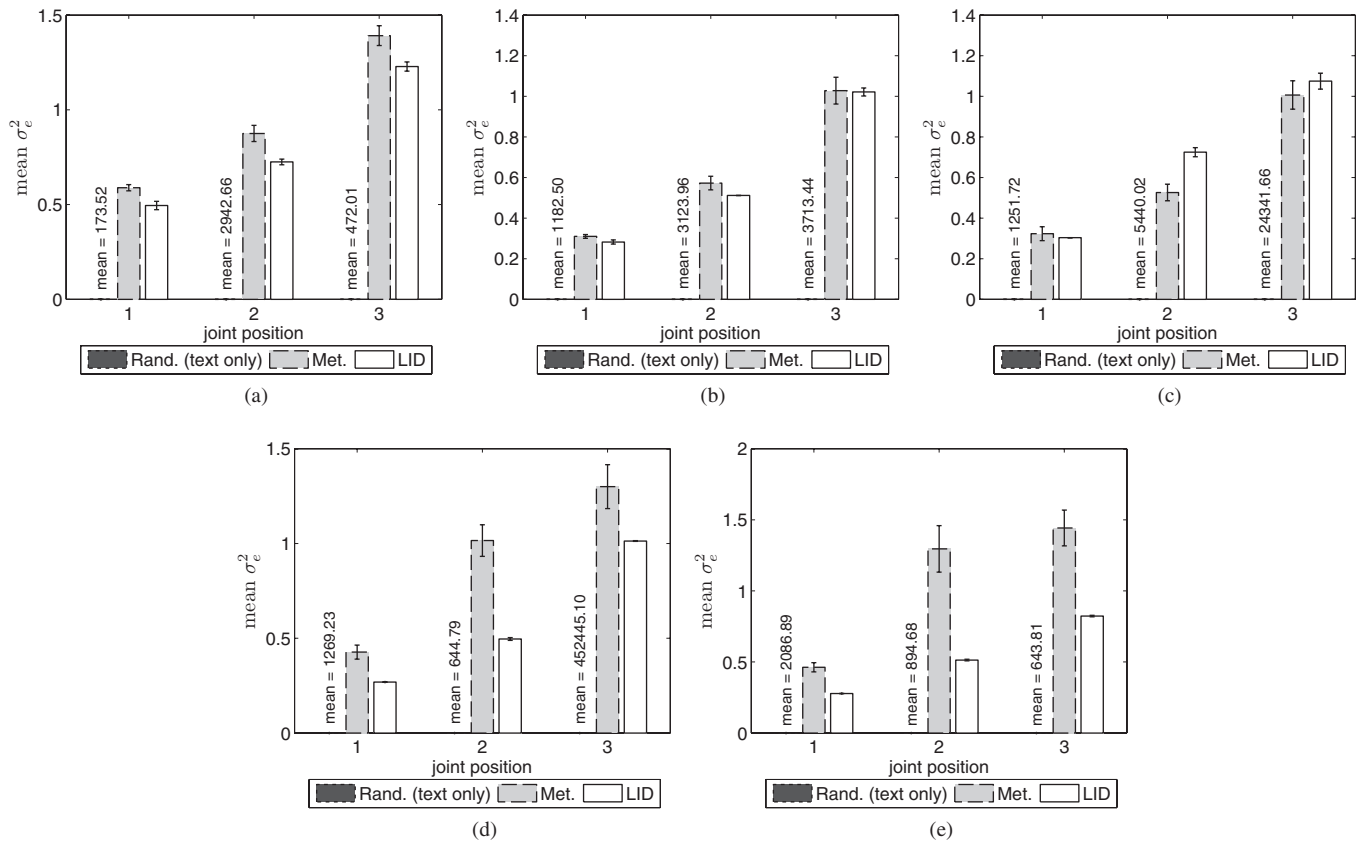


Fig. 11. Comparison of  $\sigma_e^2$ , with 95% confidence intervals shown on the bars. (a)–(e) Geometry a–e.

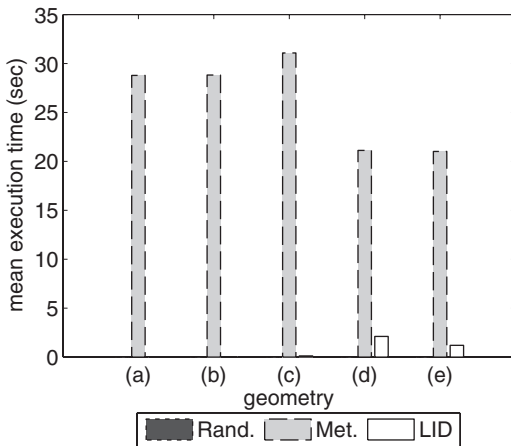


Fig. 12. Comparison of execution time. (a)–(e) Geometry a–e.

immediately after execution, but it gives poor results. The Metropolis-based method trades notably longer execution time for better results. Evidently, the LID-based method is both effective and efficient.

### C. Effects of the Number of Sensors

The previous results only consider  $m = 4$ . Here we vary  $m$  to observe the effect. For each value of  $m$ , we repeat

the previous simulations for our proposed methods. The random deployment method is ignored (for better presentation). Figs. 13 and 14 show the mean  $\alpha$  and the mean  $\sigma_e^2$ , respectively, where we denote by “J1” the joint position 1 (refer to Fig. 9), by “J2” the joint position 2, and so on. The results show that deploying more sensors does reduce the errors. The LID-based method is still preferred. Note that in Fig. 13(c), for the joint position 2, the LID-based method can outperform the Metropolis-based method by using  $m = 5$  sensors. Fig. 15 compares the execution time. The LID-based method is not sensitive to the value of  $m$ , whereas the Metropolis-based method has linearly increasing execution time as  $m$  increases.

### D. Real Gravity Measurement

In addition to simulations, we also conducted real experiments to measure the gravity vector. We constructed a rectangular box, fixed the box by a hinge joint, deployed eight lab-made sensor nodes on its surface, and then rotated the box to test the measurement. Each sensor node is equipped with a triaxial accelerometer and a triaxial gyroscope. Fig. 16 shows our experiment scenario and how we rotated the box around the beam. Here, the joint is set to be  $(0, 0, 0)$  of the  $r$ -coordinate, whose three axes are denoted by  $x_r$ ,  $y_r$ , and  $z_r$ , respectively. Note that the  $x_r$  axis is along the beam. We rotated the box “up” and “down” around the beam repeatedly. The sensors sampled the motion at a rate of 10 Hz. The

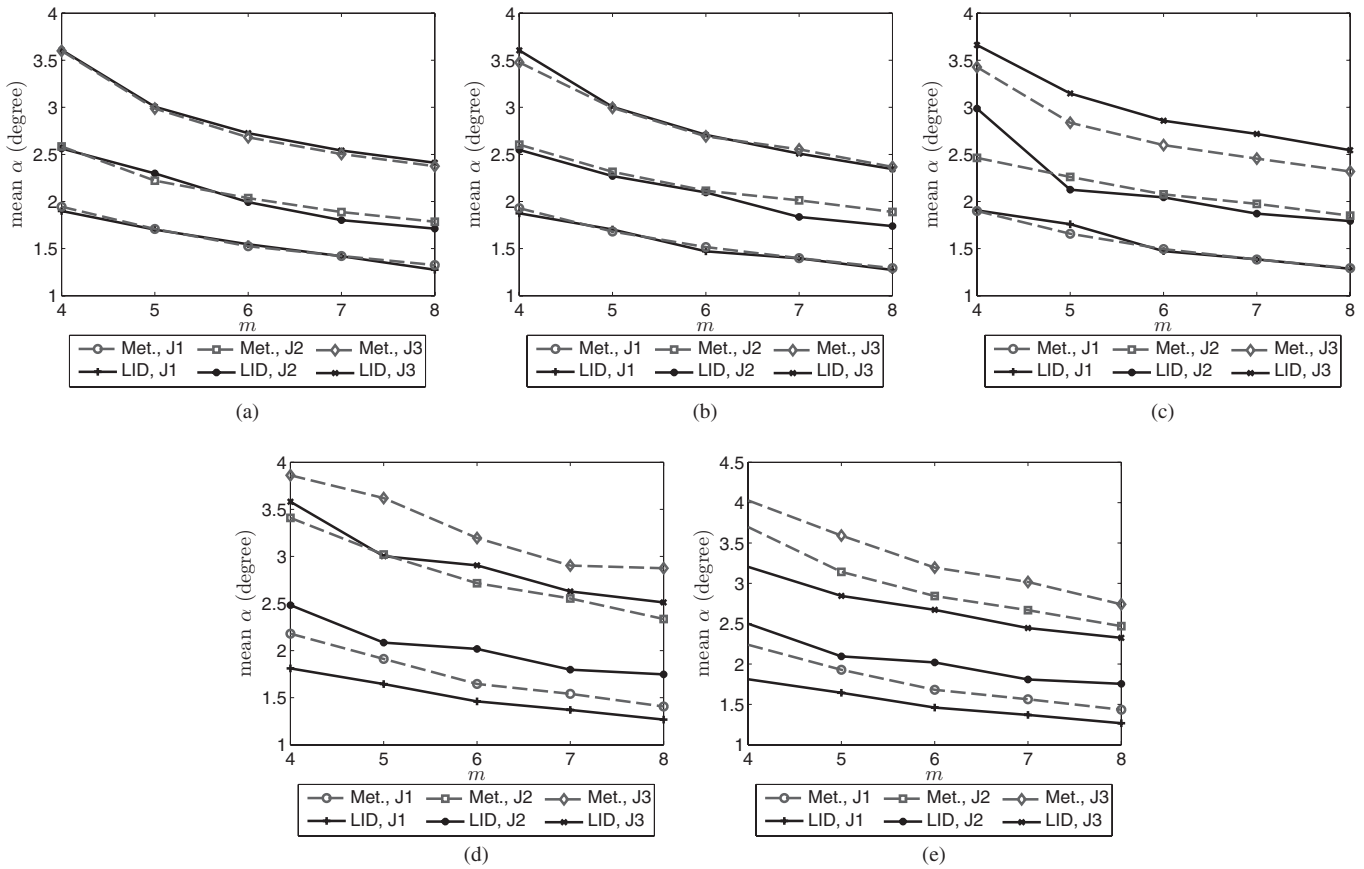


Fig. 13. Effects of  $m$  on  $\alpha$ . (a)–(e) Geometry a–e.

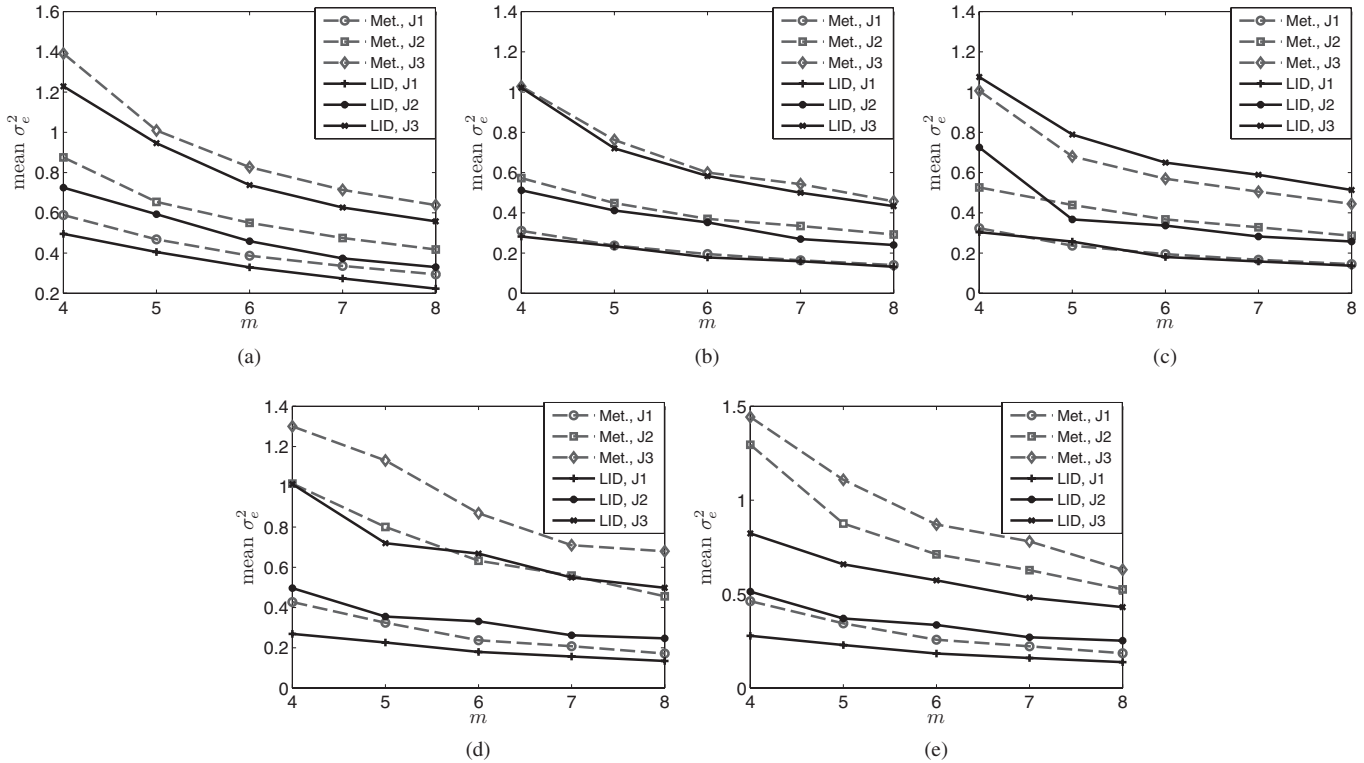


Fig. 14. Effects of  $m$  on  $\sigma_e^2(P)$ . (a)–(e) Geometry a–e.

sensing data was stored at each sensor node and reported to a sink in a reliable way. Then, for the accelerometers, we

computed the estimated gravity vector  $\hat{g}(t)$  with respect to the  $r$ -coordinate. For comparison, we also measure the rotation of



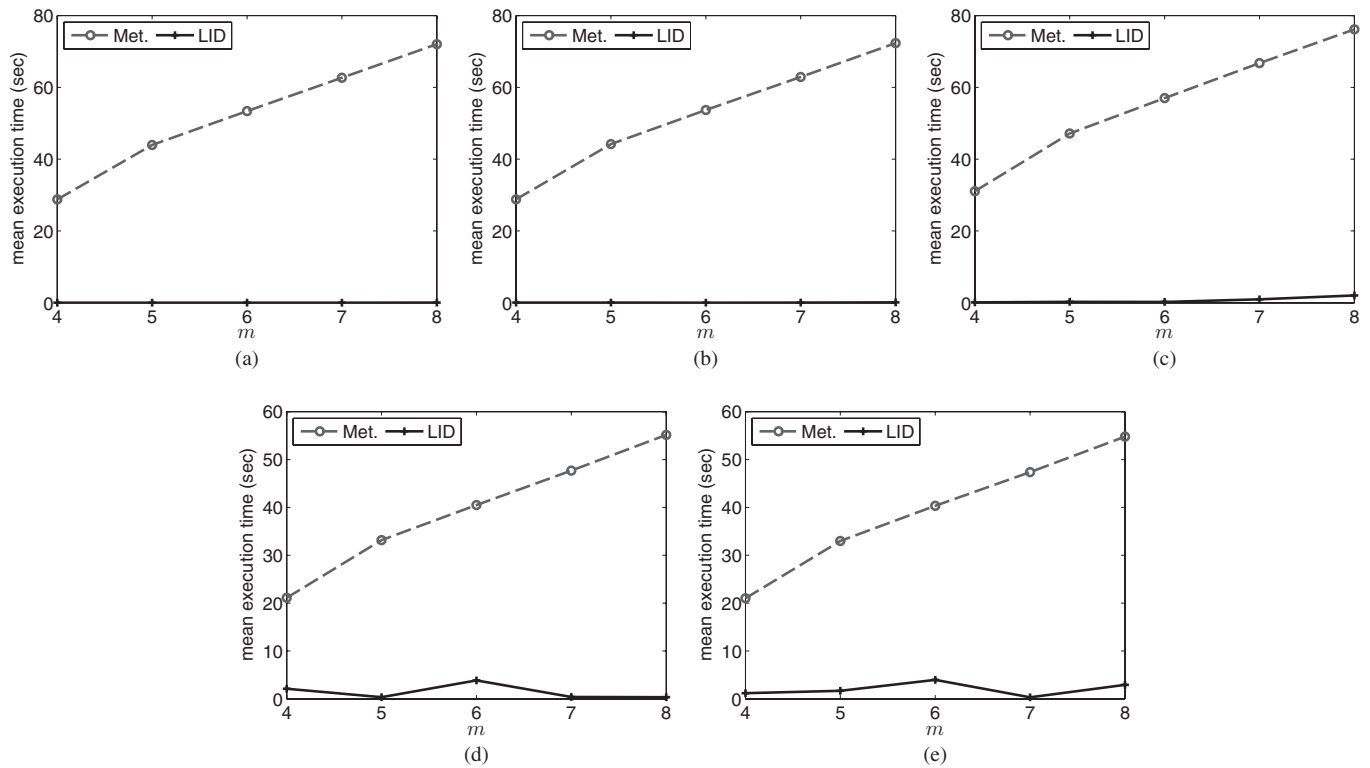


Fig. 15. Effects of  $m$  on execution time. (a)–(e) Geometry a–e.

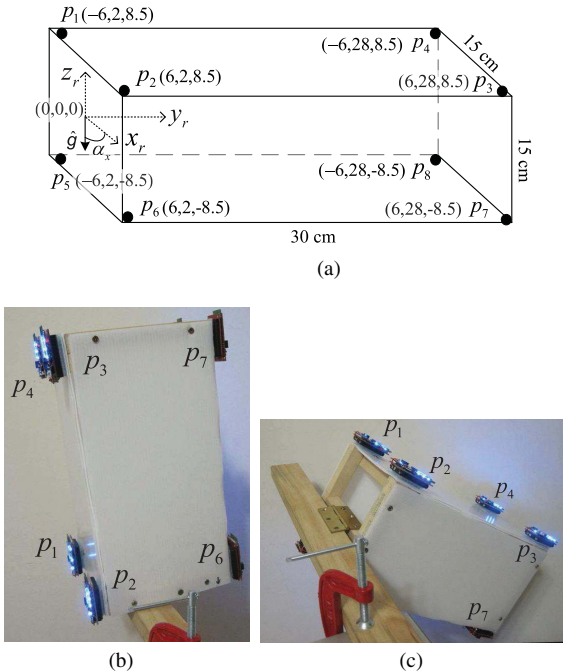


Fig. 16. Our experimental scenario (a) geometry model, (b) state “up,” and (c) state “down.”

the box by integrating the angular velocities obtained from a gyroscope.

To show the accuracy, we note that the real gravity vector  $R^T \vec{g}(t)$  is unavailable. Since any measurement method is accompanied with its own errors, it is difficult to show the

accuracy of a method by numerical comparison with other methods. However, some ground truths are available in our designed experiment. The most obvious one is that the direction of  $\hat{g}(t)$  should correspond to how we rotated the box. Let  $\alpha_x(t)$  be the angle between  $\hat{g}(t)$  and  $x_r$ , and  $\alpha_y(t)$  be that between  $\hat{g}(t)$  and  $y_r$ . We compare three schemes. Fig. 17(a) shows the result when only the accelerometer at  $p_3$  is used to measure the gravity (so,  $\hat{g}(t)$  is the raw data). It shows how the traditional scheme of a single accelerometer would perform. Fig. 17(b) shows the result of our scheme, where we use all accelerometers to compute  $\hat{g}(t)$ , which is obtained from the first column of  $\hat{Q}(t)$ . Fig. 17(c) shows the result obtained from the gyroscope at  $p_3$ . We directly measure  $\alpha_x(t)$  and  $\alpha_y(t)$  by integrating the angular velocities around  $x_r$  and  $y_r$ , respectively.

For the results in Fig. 17, ideally,  $\alpha_x(t)$  should be a constant (because the rotation was about the  $x_r$ -axis) and  $\alpha_y(t)$  should show an “up-down” pattern in the range of  $90^\circ$  to  $180^\circ$ . Since a gyroscope is less sensitive to motion accelerations, we use Fig. 17(c) as a reference to compare our scheme with the traditional scheme. Note that a gyroscope also suffers from its own problems, such as drifting biases. We only consider the tendency of its waveform to be similar to the ground truth. Now, for  $\alpha_x(t)$ , it should be evident that our scheme is closer to Fig. 17(c) whereas Fig. 17(a) is quite unstable. The  $\alpha_y(t)$  of our scheme is also closer to Fig. 17(c). The unstable peak values of Fig. 17(a) show that it suffers more from motion accelerations when the rotation slows down and begins to change direction. Our scheme is able to exploit multiple accelerometers to eliminate such disturbance. The

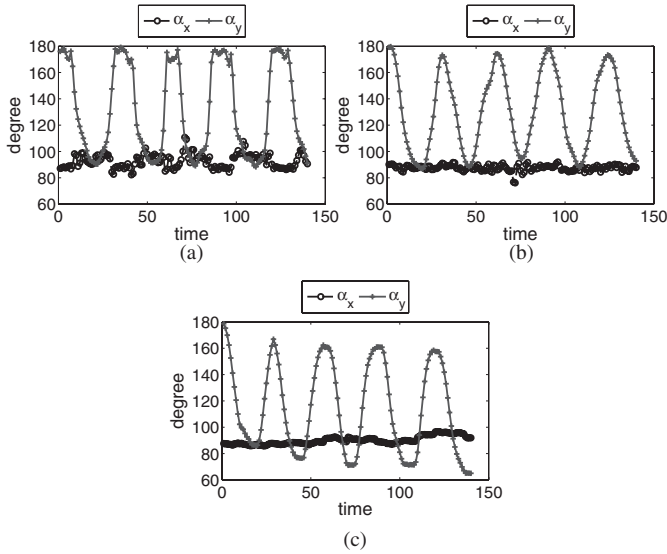


Fig. 17. Angles between  $\hat{g}(t)$  and the axes. (a) Traditional. (b) Our scheme. (c) Gyroscope.

experimental results show that our scheme works well in real situations.

## VI. CONCLUSION

In this paper, we have investigated the gravity measurement problem for rigid bodies. We have modeled the sensor deployment problem by considering the locations of sensors on one rigid body to minimize the measurement errors. The model is quite suitable for BISNs. Based on our formulation, we proposed two heuristics to find near-optimal deployments for arbitrary rigid bodies. One is based on a probabilistic optimization technique that eventually finds near-optimal solutions. The other is based on a “largest inter-distance” guideline, which leads to optimal solutions in some special cases. Experimental results showed that our methods are quite effective and significantly outperform previous approaches.

### APPENDIX I PROOF OF THEOREM 1

In Section IV-B, we show that Theorem 1 holds for a sphere of radius  $\sqrt{3}$ . Here, we show that Theorem 1 also holds for spheres of any radii. Specifically, let the position of a sensor  $s_i$ ,  $i = 1, \dots, 4$ , be  $p_i = (p_{ix}, p_{iy}, p_{iz})$ . Deploying  $s_i$  on a sphere of radius  $d$  implies that  $p_{ix}^2 + p_{iy}^2 + p_{iz}^2 = d^2$ . We show that when  $p_i$ s form a regular tetrahedron, the deployment is optimal.

A deployment matrix  $P$  is a  $4 \times 4$  matrix, where each column is a four dimensional vector. Although the geometrical interpretation of singular value decomposition (SVD) of  $P$  plays an important role in understanding Theorem 1, it is hard to show the SVD process for four dimensional vectors. Noting that the LID guideline can be applied under dimension reduction of  $P$ , instead of considering a 3D sphere and its four-dimensional  $P$  directly, we will first consider deploying  $m = 3$  sensors on a 2D circle, analyze properties of its

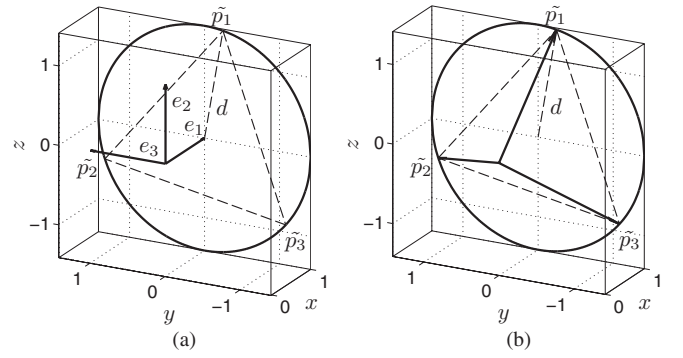


Fig. 18. SVD process for a regular triangle. (a) Before multiplying  $P$ . (b) After multiplying  $P$ .

three-dimensional  $P$  (a  $3 \times 3$  matrix) geometrically, and show that following the LID guideline, a regular triangle deployment is optimal. Then, the analysis will be extended to a four-dimensional  $P$  in a straightforward manner to complete the proof.

In the 2D reduction, three sensors are deployed on a circle of radius  $d$ . Each  $s_i$ 's location  $p_i = (p_{ix}, p_{iy})$  satisfies  $p_{ix}^2 + p_{iy}^2 = d^2$ . The deployment matrix becomes

$$P = \begin{bmatrix} 1 & 1 & 1 \\ p_{1x} & p_{2x} & p_{3x} \\ p_{1y} & p_{2y} & p_{3y} \end{bmatrix}. \quad (7)$$

Similarly,  $|P|$  has a fixed value. Below, we apply Lemma 1 to Eq. (7) to show that when  $p_i$ s form a regular triangle (the counterpart of a regular tetrahedron), the deployment is optimal.

Each column of  $P$  in Eq. (7) is a vector that points to a circle on the  $x = 1$  plane [(refer to Fig. 18(b)]. It turns out that when we consider a circle of any radius  $d$ , the equalities  $\rho_1(P) = \rho_2(P) = \rho_3(P)$  in Lemma 1 may not exist. To overcome such difficulties, we extend  $P$  in Eq. (7) to

$$\tilde{P} = \begin{bmatrix} \delta & \delta & \delta \\ p_{1x} & p_{2x} & p_{3x} \\ p_{1y} & p_{2y} & p_{3y} \end{bmatrix}$$

where  $\delta > 0$ . Note that in the 3D space of  $\tilde{P}$ , the circle is translated from the  $x = 1$  plane to the  $x = \delta$  plane.

*Lemma 2:* The equalities  $\rho_1(\tilde{P}) = \rho_2(\tilde{P}) = \rho_3(\tilde{P})$  hold if and only if  $\|p_1 - p_2\| = \|p_2 - p_3\| = \|p_3 - p_1\|$  and  $d = \sqrt{2}\delta$ .

*Proof:* Let the SVD of  $\tilde{P}$  be  $\tilde{P} = R_2 \Xi R_1$ , where  $R_1$  and  $R_2$  are rotation matrices, and  $\Xi$  is the diagonal matrix whose diagonal entries are  $\rho_1(\tilde{P})$ ,  $\rho_2(\tilde{P})$ , and  $\rho_3(\tilde{P})$ . As shown in Fig. 18, the three matrices are applied to the base vectors  $e_1 = (1, 0, 0)$ ,  $e_2 = (0, 1, 0)$ , and  $e_3 = (0, 0, 1)$  to carry out the transformation. If  $\rho_1(\tilde{P}) = \rho_2(\tilde{P}) = \rho_3(\tilde{P})$ , we have  $\tilde{P} = R_2 \Xi R_1 = \rho_1(\tilde{P}) R_2 R_1$ . That is, after multiplying  $\tilde{P}$ , the base vectors remain orthogonal to each other and each  $e_i$  is scaled by  $\rho_1(\tilde{P})$ . At the  $x = \delta$  plane, setting radius  $d = \sqrt{2}\delta$  preserves the orthogonality, which also implies that  $p_1$ ,  $p_2$ , and  $p_3$  form a regular triangle. The “only if” part can be shown similarly. ■

In moving from the  $x = \delta$  plane back to the  $x = 1$  plane, the following property holds.

*Lemma 3:* The product of singular values  $\rho_1(P)\rho_2(P)\rho_3(P)$  equals  $\rho_1(\tilde{P})\rho_2(\tilde{P})\rho_3(\tilde{P})/\delta$ .

*Proof:* By properties of SVD, the squares of singular values of a matrix  $P$ , namely  $\rho_k^2(P)$ s, are the eigenvalues of  $P^T P$ . Since the determinant of a square matrix equals the product of its eigenvalues, we have

$$\det(P^T P) = \rho_1^2(P)\rho_2^2(P)\rho_3^2(P).$$

We may decompose  $\tilde{P}$  as

$$\tilde{P} = \begin{bmatrix} \delta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} P = \Delta P.$$

By properties of determinants, we also have

$$\begin{aligned} \det(\tilde{P}^T \tilde{P}) &= \det\left[(\Delta P)^T (\Delta P)\right] \\ &= \det(\Delta P) \det(\Delta P) \\ &= \det(\Delta)^2 \det(P^T P) \\ &= \delta^2 \det(P^T P). \end{aligned}$$

Since singular values are larger than zero, we have the desired result. ■

Because the arithmetic mean  $\sum_{k=1}^3 \rho_k^2(\tilde{P})/3$  is fixed for all deployments, from Lemma 2 and the power mean inequality [34], we know that an extended  $\tilde{P}$  where sensors form a regular triangle has the largest geometric mean  $\sqrt{3}\rho_1^2(\tilde{P})\rho_2^2(\tilde{P})\rho_3^2(\tilde{P})$  and the largest harmonic mean  $3/\sum_{k=1}^3 \rho_k^{-2}(\tilde{P})$ , and they are equal to the arithmetic mean. Hence, by Lemma 3, we also know that the original  $P$  has the largest geometric mean. Now that the arithmetic mean is fixed for all deployments, we can expect that  $P$  has the largest harmonic mean, supporting our claim.

We now extend the above discussion to the original  $4 \times 4$  deployment matrix. We show that a regular tetrahedron, where every tuple of three points  $p_i$ ,  $p_j$ , and  $p_k$  forms a regular triangle, is an optimal deployment on a sphere. The extension can be done in a straightforward manner. Lemma 2 and Lemma 3 are extended as follows.

*Lemma 4:* The equalities  $\rho_1(\tilde{P}) = \rho_2(\tilde{P}) = \rho_3(\tilde{P}) = \rho_4(\tilde{P})$  hold if and only if  $d = \sqrt{3}\delta$  and  $p_1$ ,  $p_2$ ,  $p_3$ , and  $p_4$  form a regular tetrahedron.

*Lemma 5:* The following equality holds for a deployment matrix  $P$  and its extension  $\tilde{P}$ :  $\rho_1(P)\rho_2(P)\rho_3(P)\rho_4(P) = \rho_1(\tilde{P})\rho_2(\tilde{P})\rho_3(\tilde{P})\rho_4(\tilde{P})/\delta$ .

Again, since the arithmetic mean of  $\rho_k^2(P)$ s is fixed for all deployments, we can expect a tetrahedron, which has the largest geometric mean, to also have the largest harmonic mean. We show it by an example in Fig. 4. Since the harmonic mean is the reciprocal of error variance  $\sigma_e(P)$ , we conclude that a regular tetrahedron is optimal.

## APPENDIX II PROOF OF THEOREM 2

First, we show that Theorem 2 holds for the cube in Fig. 5(a). The four sensor locations  $p_1$ ,  $p_2$ ,  $p_3$ , and  $p_4$  in

Fig. 5(a) form a regular tetrahedron, and the deployment matrix  $P$  has the maximal norm  $|P|$ . Because a cube can be circumscribed by an outer sphere, the regular tetrahedron in Fig. 5(a) is indeed a rotation of the regular tetrahedron in Fig. 4(a). From Theorem 1, we know that a regular tetrahedron deployment is optimal for a fixed  $|P|$ . Then, the optimality ensues from Lemma 1. Similarly, Theorem 2 also holds for cubes of other edge lengths.

## REFERENCES

- [1] C.-H. Wu, Y.-T. Chang, and Y.-C. Tseng, "Multi-screen cyber-physical video game: An integration with body-area inertial sensor networks," in *Proc. Int. Conf. Pervas. Comput. Commun. Workshops*, 2010, pp. 832–834.
- [2] S. Trimpe and R. D'Andrea, "Accelerometer-based tilt estimation of a rigid body with only rotational degrees of freedom," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 2630–2636.
- [3] M. Sippel, A. Abduhl-Majeed, W. Kuntz, and L. Reindl, "Enhancing accuracy of an indoor radar by the implementation of a quaternion-and unscented kalman filter-based lightweight, planar, strapdown IMU," in *Proc. Eur. Navigat. Conf. Global Navigat. Satellite Syst.*, 2008, pp. 1–5.
- [4] D. T. W. Fong, J. C. Y. Wong, A. H. F. Lam, R. H. W. Lam, and W. J. Li, "A wireless motion sensing system using ADXL MEMS accelerometers for sports science applications," in *Proc. 5th World Congr. Intell. Control Autom.*, Jun. 2004, pp. 5635–5640.
- [5] L. Atallah, G. G. Jones, R. Ali, J. J. Leong, B. Lo, and G.-Z. Yang, "Observing recovery from knee-replacement surgery by using wearable sensors," in *Proc. Int. Conf. Wearable Implant. Body Sensor Netw.*, May 2011, pp. 29–34.
- [6] C.-H. Wu and Y.-C. Tseng, "Data compression by temporal and spatial correlations in a body-area sensor network: A case study in pilates motion recognition," *IEEE Trans. Mobile Comput.*, vol. 10, no. 10, pp. 1459–1472, Oct. 2011.
- [7] D. Vlastic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović, "Practical motion capture in everyday surroundings," *ACM Trans. Graph.*, vol. 26, no. 3, p. 35, Jul. 2007.
- [8] J. K. Lee and E. J. Park, "A minimum-order Kalman filter for ambulatory real-time human body orientation tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3565–3570.
- [9] J. Brutovsky and D. Novak, "Low-cost motivated rehabilitation system for post-operation exercises," in *Proc. 28th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug.–Sep. 2006, pp. 6663–6666.
- [10] C.-C. Lo, C.-P. Chiu, Y.-C. Tseng, S.-A. Chang, and L.-C. Kuo, "A walking velocity update technique for pedestrian dead-reckoning applications," in *Proc. IEEE Symp. Pers., Indoor, Mobile Radio Commun.*, Sep. 2011, pp. 1249–1253.
- [11] C. Kallmayer and E. Simon, "Large area sensor integration in textiles," in *Proc. 9th Int. Multi-Conf. Syst., Signals, Devices*, Mar. 2012, pp. 1–5.
- [12] J. Chen, E. Shen, and Y. Sun, "The deployment algorithms in wireless sensor networks: A survey," *Inf. Technol. J.*, vol. 8, no. 3, pp. 293–301, 2009.
- [13] Y.-C. Wang and Y.-C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multi-level coverage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 9, pp. 1280–1294, Sep. 2008.
- [14] C.-F. Huang, L.-C. Lo, Y.-C. Tseng, and W.-T. Chen, "Decentralized energy-conserving and coverage-preserving protocols for wireless sensor networks," *ACM Trans. Sensors Netw.*, vol. 2, no. 2, pp. 182–187, 2006.
- [15] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Trans. Sensor Netw.*, vol. 1, no. 1, pp. 36–72, 2005.
- [16] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proc. ACM Int. Workshop Wireless Sensor Netw. Appl.*, 2003, pp. 182–187.
- [17] D. Tao, H. Ma, and L. Liu, "Coverage-enhancing algorithm for directional sensor networks," in *Mobile Ad-Hoc and Sensor Networks* (Lecture Notes in Computer Science), vol. 4325. Berlin, Germany: Springer-Verlag, 2006, pp. 256–267.
- [18] Y. Wang and G. Cao, "On full-view coverage in camera sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1781–1789.

- [19] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Robust sensor placements at informative and communication-efficient locations," *ACM Trans. Sensor Netw.*, vol. 7, no. 4, pp. 31:1–31:33, 2011.
- [20] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. Leung, "Body area networks: A survey," *ACM Mobile Netw. Appl.*, vol. 16, no. 2, pp. 171–193, 2011.
- [21] G.-Z. Yang, *Body Sensor Networks*. New York: Springer-Verlag, 2006.
- [22] G. Welch and E. Foxlin, "Motion tracking: No silver bullet, but a respectable arsenal," *IEEE Comput. Graph. Appl.*, vol. 22, no. 6, pp. 24–38, Nov.–Dec. 2002.
- [23] D. A. Rodríguez-Silva, F. Gil-Castiñeira, F. J. González-Castaño, R. J. Duro, F. López-Peña, and J. Vales-Alonso, "Human motion tracking and gait analysis: A brief review of current sensing systems and integration with intelligent environments," in *Proc. IEEE Int. Conf. Virtual Environ., Human-Comput. Inter., Meas. Syst.*, Jul. 2008, pp. 166–171.
- [24] B. Huyghe, J. Dautreloigne, and J. Vanfleteren, "3D orientation tracking based on unscented kalman filtering of accelerometer and magnetometer data," in *Proc. IEEE Sensors Appl. Symp.*, Feb. 2009, pp. 148–152.
- [25] Y.-X. Lai, L. Shu, A. Vasilakos, J. J. P. C. Rodrigues, C.-F. Lai, and Y.-M. Huang, "3D adaptive reconstruction of human motion from multi-sensors," in *Proc. 3rd Int. Workshop Wireless Sensor, Actuat., Robot Netw.*, 2011, pp. 531–536.
- [26] A. D. Young, "Use of body model constraints to improve accuracy of inertial motion capture," in *Proc. Int. Conf. Wearable Implantable Body Sensor Netw.*, Jun. 2010, pp. 180–186.
- [27] Z. Zhang, L. W. Wong, and J.-K. Wu, "3D upper limb motion modeling and estimation using wearable micro-sensors," in *Proc. Int. Conf. Wearable Implantable Body Sensor Netw.*, Jun. 2010, pp. 117–123.
- [28] Z. Zhang, "Ubiquitous human motion capture using wearable micro-sensors," in *Proc. Int. Conf. Pervas. Comput. Commun.*, Mar. 2009, pp. 1–2.
- [29] J. Carig, *Introduction to Robotics*. Englewood Cliffs, NJ: Prentice-Hall, 2005.
- [30] A. D. Young, "Comparison of orientation filter algorithms for realtime wireless inertial posture tracking," in *Proc. 6th Int. Conf. Wearable Implantable Body Sensor Netw.*, Jun. 2009, pp. 59–64.
- [31] D. R. Basu and A. Lazaridi, "Stochastic optimal control by pseudo-inverse," *Rev. Econ. Stat.*, vol. 65, no. 2, pp. 347–350, 1983.
- [32] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *Amer. Stat.*, vol. 49, no. 4, pp. 327–335, 1995.
- [33] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
- [34] J. M. Steele, *The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [35] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 640–652, Jun. 2006.
- [36] *Unity—3D Game Engine*. (2011) [Online]. Available: <http://unity3d.com>



**Chun-Hao Wu** received the Ph.D. degree in computer science from National Chiao-Tung University, Hsinchu, Taiwan, in 2012.

His current research interests include mobile computing, wireless communication, wireless body-area sensor networks, and parallel and distributed computing.



**Yu-Chee Tseng** received the Ph.D. degree in computer and information science from the Ohio State University, Columbus, in 1994.

He was the Chairman with the Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan, from 2005 to 2009, where he has been the Chair Professor since 2011, and the Dean since 2011. He is a Y. Z. Hsu Scientific Chair Professor. His current research interests include mobile computing, wireless communication, and sensor networks.

Dr. Tseng was a recipient of the Outstanding Research Award from the National Science Council in 2001, 2003, and 2009, the Best Paper Award at the International Conference on Parallel Processing in 2003, the Elite I. T. Award in 2004, the Distinguished Alumnus Award from Ohio State University in 2005, and the Y. Z. Hsu Scientific Paper Award in 2009. He was on the Editorial Board of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2005 to 2009 and the IEEE TRANSACTIONS ON MOBILE COMPUTING from 2006 to 2011. He has been on the Editorial Board of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS since 2008.