

# Beyond One-dollar Mouse: A Battery-free Device for 3D Human-Computer Interaction via RFID Tags

Qiongzhen Lin\*, Lei Yang\*, Yuxin Sun\*, Tianci Liu\*, Xiang-Yang Li\*<sup>†</sup>, Yunhao Liu\*

\* School of Software and TNLIST, Tsinghua University, China

<sup>†</sup> Department of Computer Science, Illinois Institute of Technology, USA

E-mail: {lin,young,sun,tian}@tagsys.org,xiangyang.li@gmail.com, yunhao@greenorbs.com

**Abstract**—For today’s computer users, the mouse plays such an important role that it dominates the interaction interface in personal computer for nearly half a century since it was invented. However, the mouse is gradually unfit for the demand of modern 3D display techniques, *e.g.* 3D-projection or -screen, for the reason that the relevant interactions are confined in a surface. Although some new methods such as computer vision based techniques attempt to bridge the human-computer barrier, they suffer from many limitations such as ambiguity in multi-targets and dependence on light. This paper presents a battery-free device called *Tagball* for 3D human-computer interaction via RFID tags. *Tagball* devises a control ball, on which  $N$  passive tags are attached, for users to generate two basic kinds of interactive commands: translation and rotation. Instead of locating  $N$  tags independently, we model the ball as a whole in a more cooperative way under the circumstance that their geometric relationships are known in advance. In addition, we consider the phase values measured by  $M$  RF antennas for these  $N$  tags as observations of the ball state. Our key innovations are the studies on motion behaviors of a group of tags by using Extended Kalman Filter, and the implementation based on purely Commercial Off-The-Shelf (COTS) RFID products. The systematical evaluation shows that *Tagball* traces the ball translation to  $1.5cm$  and identifies ball orientation to  $1.8^\circ$  in 3D space.

**Keywords**—RFID, *Tagball*, Interaction Peripheral, Extended Kalman Filter

## I. INTRODUCTION

Radio-frequency identification (RFID) technique has been increasingly used in everyday scenarios, ranging from warehouse inventory to tracking [1]. The reason for this widespread deployment is the simplicity of tags, which enables very low cost at high volumes. A tag has small microchips and an antenna on board. The readers can collect the IDs of tags via RF signals, without keeping them in sight or within reach. The initial motivation of RFID is to automatically identify objects, but its potential applications have been widely studied in various areas in recent years. In this paper, we are going to discuss its application in human-computer interaction.

The mouse which was invented by Douglas Engelbart in 1963, dominates the interaction interface in personal computer for nearly half a century. It is a pointing device that detects two-dimensional motion relative to a surface and plays an important role for today’s computer user in 2D world. However, the mouse is gradually unfit for the demand of modern 3D display techniques, such as 3D-projection or -screen. For example, a user has to move the mouse along a big circle to perform a simple rotation action.

There is growing interest in the development of new approaches and technologies for bridging the human-computer barrier and facilitating natural human computer interaction in 3D world. They can be grouped into three categories: (i) Computer vision based systems, such as Kinect, Wii, LeapMotion, make use of depth sensors or infrared cameras to allow a user to interact with computers. These methods look fantastic and are close to human habit, but they almost inherit every shortcoming from the computer vision, such as dependence on light, dead corner, high computation cost, and ambiguity in multi-targets. (ii) Sensor based systems, like using data glove or smart phone, are not designed for improvement on interaction with personal computers. (iii) device-free systems [2], [3] appear attractive but are confined in body-level.

Advances in semiconductor technology in recent years promote the rapid development of RFID technique. Nowadays RFID reader is becoming a common module integrated with a variety of consumer electronics *e.g.* smartphone, tablet, topset, *etc.* In this paper, we leverage passive RFID tags to design a novel human interaction device called *Tagball* to extend the function of mouse. The overall design idea is very simple: a group of tags are attached on a control ball with known geometric relationships. Like using mouse, a user translates or rotates the ball to adjust the visual field, browse graphic model, or even control the remote telemedicine devices. The *Tagball*, devoted to 3D interaction, overcomes mouse’s shortcomings while retaining its simplicity and usability. It does not contain any additional sensing components, thereby, its cost is less than 1 US dollar<sup>1</sup>. Importantly, it is battery free so that the user does not need to worry about battery exhaustion.

Our goal is to design and build a system that traces the ball state (referring to its position and orientation) through  $N$  tags, which are attached on the ball. The naive method is to localize these  $N$  tags independently by using  $M$  RF antennas, then to calculate the ball state through their positions. Even though pinpointing or tracking individual tag has been widely studied [4]–[7] in recent years, this naive method is infeasible in our scenario for three reasons. First, phase measurements are not accurate; instead they contain random errors, following a Gaussian distribution [6]. What’s worse, Doppler effect occurring in the mobile tags further aggravates the problem. Second, the ball should be designed with smaller size to offer a user-friendly device, while it is hard to obtain positions of tags with dense deployment in a small space. Third, it is known that water is a killer factor that hinders the

<sup>1</sup>Suppose RF reader has been integrated at the end machine.

UHF signal propagation. Water-rich human palm may obstruct the reading of parts of tags.

Instead of viewing  $N$  tags independently, we consider these tags as a whole in a more cooperative way under the circumstance that their geometric relationships are already known. To deal with the challenges above, we model the ball as a perfect rigid body and parameterize its state by using Roll-Pitch-Yaw  $\Phi(\phi_x, \phi_y, \phi_z)$  plus the position  $\mathcal{O}(x_0, y_0, z_0)$ . Correspondingly, the phase measurements measured by  $M$  antennas are viewed as the observations of the ball state. This approach is attractive in many aspects. (i) it does not make any assumption about each tag's motion, which makes it is capable of handling variable number of moving tags. (ii) it puts a constraint on availability of phase measurement through the tags' geometric relationships. (iii) there are many sophisticated filter techniques. They use a series of observations which contain noise and other inaccuracies over time to produce a statistically optimal estimation system state. We design three main components converting the RF signals to interactive commands. First, the initial ball state is estimated by using a correlation method, which is further converted into an optimization problem (see Section IV). Second, we combine the spatial- and time-domain positioning information by utilizing a dynamic model of Extend Kalman Filter for the ball (see Section V), which cannot be solved via using any single positioning method. At last, we use a backward recursion to smooth the states (see Section VI). In summary, our work makes three key contributions:

- First, to the best of our knowledge, Tagball is the first system that studies motion behaviors of a group of tags with known geometric relationships. It offers a novel way of human-computer interaction in 3D space, which overcomes the mouse's limitations while inheriting its simplicity and usability.
- Second, we view the ball as a whole and the phase measurements as the state observations. Then the technique of Extended Kalman Filter is introduced to trace the ball state, allowing part of tags not to be read and tolerating inaccurate phase measurements.
- Third, Tagball is designed and implemented purely based on COTS RFID products, which makes the fast adoption and deployment possible. The systematical evaluation shows that it traces the ball location to  $1.5cm$  and identifies ball orientation to  $1.8^\circ$ .

The rest of this paper is organized as follows. Section II reviews the related work. The system model and architecture are presented in Section III. We present the details of Tagball design in Section IV, Section V and Section VI. A proof-of-concept prototype of Tagball is implemented and evaluated in Section VII. Finally we make the conclusion in Section VIII.

## II. RELATED WORK

This section reviews the related work including the novel interactive devices and RFID based localization.

### A. Novel interactive device

There are many novel interactive devices, which can be classified into four categories:

**Computer vision based methods:** Another works leverage various cameras to capture and recognize the human's gesture. Typically, the Microsoft Kinect is one of the pioneers who use computer vision technique to recognize user's joints, motion and gestures. LeapMotion is another developing motion sensing technology invented in 2010. It is designed to track user's fingers (or items such as a pen) which cross to the observed area. These methods almost inherit every shortcoming from the computer vision. Different from the imaging or infrared based solutions mentioned above, Tagball is an RF-based device, which does not require all tags in line-of-sight. What's more, since tags have unique IDs, easily-extensible to allow a large number of users to interact with the system simultaneously without causing confusion.

**Sensor based methods:** These works [8], [9] equip a data glove with various kinds of sensors. This glove has the capability of recording hand movements, positioning hand and its orientation as well as finger movements. The main advantage of data glove is its high sensitivity which is suitable for particular area, like telemedicine and film-making. Nevertheless, it cannot be applied in personal computer in a short span of time due to its high cost. By holding the phone like a pen, the user is able to write short message or draw simple diagrams in the air [10]. [11] identifies the location of screen taps on smartphones and tablets from accelerometer and gyroscope readings. Most of these methods are used for improvement the interaction with smart phones instead of PC.

**Device-free methods:** [2] uses the human body as an antenna to sense whole-body gestures. [3] is a novel gesture recognition system that leverages wireless signals to enable whole-home sensing and the recognition of human gestures. [12] designs a gesture-recognition system that can operate on a range of computing devices including those with no batteries. SoundWave [13] utilizes the speaker and microphone already embedded in most commodity devices to sense in-air gestures around the PC. Device-free methods appear attractive and relevant to us. However, their control is body-level and they cannot be used for accurate and fine-grained gestures.

**RF based methods:** [14] presents a system that traces the trajectory of RF source. The system enables a virtual touch screen based on RF signals. However, it only works on a surface. Instead, our system totally supports rotations and translations in 3D space.

### B. RFID based localization

There is a growing interest in using phase information to locate tags. These methods can be divided into two groups, AoA (Angle of Arrival) and SAR (synthetic aperture Radar). AoA locates the tag by measuring the phase difference between the received signals at different antennas [15], [16]. The major challenge for these methods is how to deal with NLOS. State-of-the-art systems use SAR for object localization and terrain imaging with the help of antenna array [4], [6]. PinIt [4] utilizes the multipath profile to find the nearest reference tags. [5] extends this technique to robot object manipulation. Our system is inspired by the recent work [4]–[6], which study the RFID based localization or trajectory tracking with high precision. A key difference between their work and ours is that we focus on studying the behaviors of a group of tags instead of the behaviors of individuals in a small space.

TABLE I: Terms used in the description of Tagball

Term	Definition
<b>W</b> and <b>B</b> $M$ and $N$	The world coordinate system and ball coordinate system expressing two kinds of locations. The total numbers of reader antennas and tags.
$\mathcal{O}(x_0, y_0, z_0)$ $\Phi(\phi_x, \phi_y, \phi_z)$ $\mathbf{S} = [\mathcal{O} \ \Phi]^T$	The origin of ball coordinate system (center of ball) expressed in world coordinate system. The orientation of the ball in ball coordinate system. The ball state including its position and orientation.
$T_1, \dots, T_N$ $T_1(x_1^w, y_1^w, z_1^w)^T \dots T_N(x_N^w, y_N^w, z_N^w)^T$ $T_1(x_1^b, y_1^b, z_1^b)^T \dots T_N(x_N^b, y_N^b, z_N^b)^T$	The $N$ passive tags deployed on the ball. The locations of $T_1, \dots, T_N$ expressed in world coordinate system. The locations of $T_1, \dots, T_N$ expressed in ball coordinate system.
$A_1, \dots, A_M$ $A_1^w(x_1^w, y_1^w, z_1^w)^T, \dots, A_M^w(x_M^w, y_M^w, z_M^w)^T$	The $M$ RF antennas connected to the reader. The locations of $A_1, \dots, A_M$ in the world coordinate systems
$\theta_{m,n}$ and $\hat{\theta}_{m,n}$ $e^{j(\theta_{m,n} - \hat{\theta}_{m,n})}$	The theoretical and measured phase value detected by the $m^{\text{th}}$ RF antenna and backscattered from $n^{\text{th}}$ tag. The complex exponential signal correlated with a pair of theoretical and measure phase value.
$\mathbf{X}(t), \mathbf{Z}(t), \hat{\mathbf{X}}(t),$ and $\hat{\mathbf{Z}}$ $f(\cdot)$ and $h(\cdot)$	Process state vector, observation vector, estimated state and observation. Process nonlinear vector function and observation nonlinear vector function.

### III. SYSTEM DESIGN

Tagball is a low-cost RFID-enabled solution on offering a novel human-computer interaction in 3D space. In this section, we present the system model and architecture.

#### A. System Model

The Tagball is composed of two simple devices, *control ball* and *RF receiver*, which are respectively modelled as below. For reference, all terms used in this paper are defined in Table I.

**Modeling control ball:** The control ball is a common plastic ball on which  $N$  UHF passive tags are attached, as shown in Fig. 1-①. The ball is a perfect 3D rigid body that all points on its surface maintain the distance related to each other. The motion of control ball in a reference space has six degrees of freedom. There are many ways to parameterize the motion of rigid body. Taking the symmetry of the ball into consideration, we choose the Roll-Pitch-Yaw expression  $\Phi(\phi_x, \phi_y, \phi_z)$  where the three angles respectively indicate the rotation of ‘roll’, ‘yaw’ and ‘pitch’. In addition to its position  $\mathcal{O} = (x_0, y_0, z_0)$  (defined as the location at ball’s centeriod), the ball state can be parameterized as follows:

$$\mathbf{S} = [\mathcal{O} \ \Phi]^T = [x_0 \ y_0 \ z_0 \ \phi_x \ \phi_y \ \phi_z]^T \quad (1)$$

The superscript  $T$  means the matrix or vector transpose. The ultimate goal of Tagball is to trace the changes of ball state, which are further converted to the client commands.

**Modeling coordinate system:** There are two kinds of coordinate systems that express locations, world coordinate system **W** and ball coordinate system **B**. The system **W** is the reference coordinate for points in the workspace built on the geometric relationships among reader antennas, whose origin is at the center of the RF antenna locations. The system **B** is the coordinate that points of the ball are defined, whose origin is at the center of the ball. We use  $T_n^w(x_n^w, y_n^w, z_n^w)^T$  and  $T_n^b(x_n^b, y_n^b, z_n^b)^T$  to denote the  $n^{\text{th}}$  tags’ coordinates in the system **W** and **B** respectively. The tags’ locations in system **B** are known in advance. Given the ball state  $\mathbf{S}$ , the locations in two coordinate systems can be transferred as follows.

$$T_n^w = \mathcal{R} \cdot T_n^b + \mathcal{O} \quad (2)$$

where  $\mathcal{R}$  is the rotation matrix. According to the Euler formulation,  $\mathcal{R}$  is calculated, using  $\Phi$ , as follows.

$$\begin{aligned} \mathcal{R} &= \text{RPY}(\Phi) \triangleq \text{rot}(z, \phi_z) \text{rot}(y, \phi_y) \text{rot}(x, \phi_x) \\ &= \begin{bmatrix} c_z c_y & c_z s_y s_x - s_z c_x & c_z s_y c_x + s_z s_x \\ s_z c_y & s_z s_y s_x + c_z c_x & s_z s_y c_x - c_z s_x \\ -s_y & c_y s_x & c_y c_x \end{bmatrix} \end{aligned} \quad (3)$$

where  $s_x = \sin(\phi_x)$ ,  $c_x = \cos(\phi_x)$ ,  $s_y = \sin(\phi_y)$ ,  $c_y = \cos(\phi_y)$ ,  $s_z = \sin(\phi_z)$ , and  $c_z = \cos(\phi_z)$ . The operator  $\text{rot}(\text{axis}, \text{angle})$  denotes the rotation with the specified angle around the specified axis. Above equation can be understood as follows. Any rotation  $\text{RPY}(\Phi)$  can be achieved by composing three elemental rotations with  $z$ ,  $y$ , and  $x$  axis. Totally, Eqn. 2 means the tag firstly rotates along the three axes in the ball coordinate system and then moves as the ball’s translation to position  $\mathcal{O}$ . Motion dynamics [17] indicates the relationships of rotation and translation between two consecutive time which can be expressed as:

$$\mathcal{O}(t+1) = \mathcal{O}(t) + \Delta\mathcal{O}(t) \quad (4)$$

$$\mathcal{R}(t+1) = \mathcal{R}(t) \cdot \Delta\mathcal{R}(t) \quad (5)$$

where  $\Delta\mathcal{R}$  is the incremental rotation matrix produced by the rotational motion between time  $t$  and  $t+1$ , and  $\Delta\mathcal{O}$  is the translational vector within that time period. If the rotational and translational velocities are constants,  $\Delta\mathcal{O}(t)$  and  $\Delta\mathcal{R}(t)$  are also constants.

**Modelling RF receiver:** The RF receiver, connected to the client computer through wired network shown in Fig. 1-②, constitutes of a UHF reader with  $M$  reader antennas ( $M \geq 3$ ). For brevity, we use  $A_m^w$  to indicate the  $m^{\text{th}}$  antenna as well as its coordinate in system **W**. The locations of all antennas are supposed to be known in advance. The passive tags have no battery and harvest energy from the reader’s signals. They employ the backscatter communications to modulate their information to reader. The signal traverses a double distance back and forth between the reader and tag. The total phase rotation measured by the  $m^{\text{th}}$  antenna about  $n^{\text{th}}$  tag equals:

$$\theta(A_m^w, T_n^w) = \left( \frac{4\pi}{\lambda} \times |A_m^w T_n^w| \right) \bmod 2\pi \quad (6)$$

where  $\lambda$ ,  $|\cdot|$  and  $\bmod$  are the wavelength, Euclidean distance and mod operation. The phase is a periodic function with period  $2\pi$  radians which repeats every  $\frac{\lambda}{2}$  in the distance. Note that we need to perform a set of calibration experiments at first

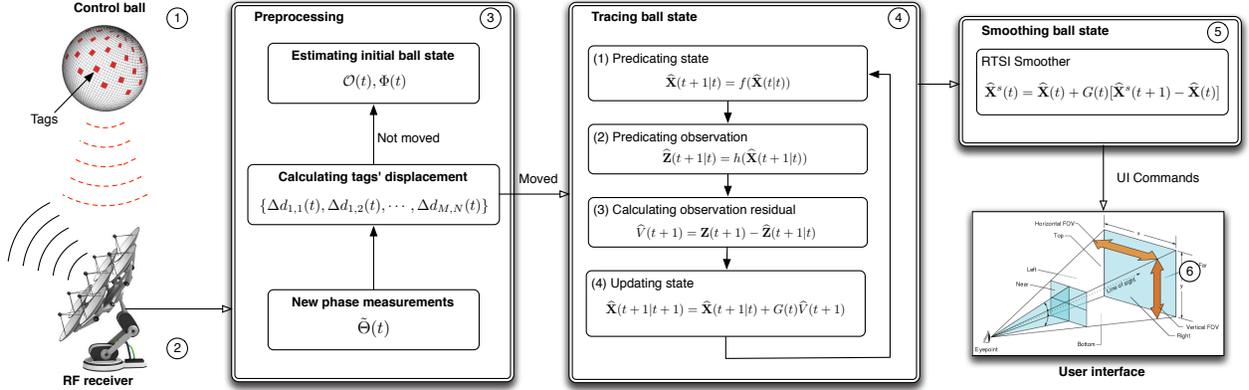


Fig. 1: System architecture

to correct the measured phase value for each pair of antenna and tag to eliminate the additional phase shift caused by the hardware's characteristics in practice [6].

To avoid the reader collisions, the antennas are alternatively scheduled in exclusive time-slots. After the  $t^{\text{th}}$  round ends, the reader outputs  $M \times N$  phase measurements. Formally, we use the  $\tilde{\Theta}(t) = \{\tilde{\theta}_{1,1}(t), \dots, \tilde{\theta}_{M,N}(t)\}$  to denote these measurements at time  $t$  where  $\tilde{\theta}_{m,n}(t)$  is the phase value measured by the  $m^{\text{th}}$  antenna for the  $n^{\text{th}}$  tag. If the reading of the  $n^{\text{th}}$  tag is missed, its value is labeled null using term  $\perp$ . Since our algorithm does not require all tags being read, the process of null value is ignored. We sometimes omit the time parameter for brevity and think that the value is inferred at time  $t$  by default.

## B. System Architecture

Fig. 1 shows the system architecture. Tagball devises three main components that convert the raw RF signals to UI commands, which can be represented as follow:

- **Preprocessing**: Tagball repeatedly reads the tags to measure the phase values. If the ball is detected as being stationary, its initial state is estimated by using a correlation method. Otherwise, the tag's displacements are calculated and fed into the next component (see Section IV).

- **Tracing ball state**: The technique of Extended Kalman Filter (EKF) is introduced to trace the ball state. The main task in this component is to define the transition and observation functions and to determine the corresponding parameters (see Section V).

- **Smoothing ball state**: The EKF only computes causal state estimates that conditioned on the previous and current observations, not on the ones obtained with a time window. The states are smoothed by using a backward recursion (see Section VI).

We describe these three components in detail in the following sections.

## IV. PREPROCESSING

The section introduces the two tasks in preprocessing: 1) estimating initial ball state; 2) calculating tag displacements.

### A. Estimating Initial Ball State

To achieve the fast state tracing, Tagball initiates a process to estimate the ball state at the beginning or every time when the control ball stops moving<sup>2</sup>. Given  $M \times N$  measured phase values,  $\tilde{\Theta} = \{\tilde{\theta}_{1,1}, \dots, \tilde{\theta}_{M,N}\}$ , our task here is to find an appropriate ball state  $\mathbf{S}$  that produces  $\tilde{\Theta}$ .

Assuming a state  $\mathbf{S}$ , we can calculate all tags' locations with Eqn. 2 and then obtain their theoretical phase values  $\Theta = \{\theta_{1,1}, \dots, \theta_{M,N}\}$  with Eqn. 6. Being similar to the augmented hologram proposed in [6]<sup>3</sup>, we define a correlation function to exhibit the likelihood how the assumed ball state  $\mathbf{S}$  is likely to be the ground truth, as below.

$$\text{Cov}(\tilde{\Theta}, \Theta) = \left| \sum_{m=1}^M \sum_{n=1}^N 2F(|\theta_{m,n} - \tilde{\theta}_{m,n}|; 0, 0.1) e^{\mathbf{J}(\theta_{m,n} - \tilde{\theta}_{m,n})} \right|$$

where  $F(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt$ . The term  $\mathbf{J}$  denotes the imaginary number and the term  $e^{\mathbf{J}\theta}$  represents a complex exponential signal. The  $F(x; \mu, \sigma)$ , the cumulative probability function of Gaussian distribution  $\mathcal{N}(\mu, \sigma)$ , is used to augment the signal amplitude whose measured value is close to theoretical one. A key point here is that all measured phase values are viewed as originating from the positions related to a same ball state. If the state  $\mathbf{S}$  is correct, the theoretical phase value will equal to the measured one. The vector of the signal  $e^{\mathbf{J}(\theta_{m,n} - \tilde{\theta}_{m,n})}$  will reach its maximum as  $\theta_{m,n} - \tilde{\theta}_{m,n}$  approaches 0. All observations from different pairs of antenna and tag constructively add up for each other. Otherwise, when  $\mathbf{S}$  is not the correct state,  $\theta_{m,n} - \tilde{\theta}_{m,n}$  will take a 'random' value in  $[0, 2\pi]$ . Different signal  $e^{\mathbf{J}(\theta_{m,n} - \tilde{\theta}_{m,n})}$  will cancel each other, as a result the final superimposing of these values at a low level.

Thus, we can convert the estimation issue to an optimization problem that finds an optimal ball state  $\mathbf{S}$  to achieve the maximum likelihood given  $\tilde{\Theta}$ , formalized as follows:

$$\max_{\mathbf{S}} \text{Cov}(\Theta, \tilde{\Theta}) \quad (7)$$

<sup>2</sup>The control ball is considered to stop moving as majority of tags' phase changes follow within a small threshold.

<sup>3</sup>We do not use the differential augmented hologram because the phase value has been calibrated.

The range of ball state is discretized in practice. It appears a large state space (6 dimensions) is needed to traverse to seek the optimized state. In fact, a very coarse initial state estimation is sufficient here, because our tracing algorithm can rapidly converge to the correct value.

### B. Calculating Tag Displacement

The nowadays COTS RFID readers have a highly efficient anti-collision algorithm which interrogate a tag every 33ms on average. Following the existing works [6] and [18], we also assume that the displacement that a tag takes every two consecutive reads is lower than  $\lambda/2 \approx 0.16m$ . Thus, we have a speed limitation that motion speed must be lower than  $\approx 5.33m/s$ . This constraint is acceptable for the majority of normal interactive tasks and can be relaxed when using more efficient anti-collision protocols. Based on Eqn. 6, the phase difference,  $\Delta\theta = \theta(t) - \theta(t-1)$ , for a pair of reader and tag between time  $t$  and  $t-1$ , relates to the difference in their distances,  $\Delta d = d(t) - d(t-1)$ , as follows.

$$\Delta\theta = \frac{4\pi}{\lambda}\Delta d + 2k\pi \quad (8)$$

where  $k$  is an integer. After simple transformation, we get  $\Delta d = \frac{\lambda}{4\pi}(\Delta\theta - 2k\pi)$ . Due to the speed constraint, the value of  $k$  should be selected from  $\{-1, 0, 1\}$ . In detail, according to the value of  $\Delta\theta$  in a particular instance, we have

$$\Delta d = \begin{cases} \frac{\lambda}{4\pi}\Delta\theta, & |\Delta\theta| \leq \pi, k = 0 \\ \frac{\lambda}{4\pi}(\Delta\theta + 2\pi), & \Delta\theta > \pi, k = -1 \\ \frac{\lambda}{4\pi}(\Delta\theta - 2\pi), & \Delta\theta < -\pi, k = 1 \end{cases} \quad (9)$$

In this way, we can estimate the motion displacements  $\Delta d_{m,n}(t)$  for each pair of antenna  $A_m$  and tag  $T_n$  after the  $t^{th}$  round schedule ends. These calculated displacements will be fed into the next component as the observations of ball's motion state.

## V. TRACING BALL STATE

Although we propose an approach to estimate ball state in previous section, it is not appropriate for real-time and accurate tracing due to huge computation. Instead, we consider the control ball as a rigid body and trace its motion state by using the Extended Kalman filter (EKF). This section firstly presents the background of EKF and then introduces how EKF is applied in our scenario.

### A. Extended Kalman Filter

Despite the existence of more sophisticated filters, Kalman filtering has been used successfully in different prediction applications or state determination of a system. Following the common practice, we adopt the Extended Kalman Filter (EKF) here.

The EKF addresses the general problem of estimating the state  $\mathbf{X}$  of a non-linear discrete-time controlled process that is governed by the stochastic difference equation:

$$\mathbf{X}(t+1) = f(\mathbf{X}(t)) + W(t) \quad (10)$$

with an observation  $\mathbf{Z}$  that is

$$\mathbf{Z}(t+1) = h(\mathbf{X}(t+1)) + V(t+1) \quad (11)$$

The random variables  $W(t)$  and  $V(t)$  represent the process and measurement noise (respectively). They are assumed to be independent from each other with Gaussian distributions. *i.e.*  $W(t) \sim \mathcal{N}(0, Q(t))$  and  $V(t) \sim \mathcal{N}(0, R(t))$ . The  $f(\cdot)$ , called *transition function*, in the Eqn. 10 relates the state at current time step  $t$  to the next  $t+1$ , in the absence of a process noise and without control-input model. The  $h(\cdot)$ , called *observation function*, represents how to obtain observations from a state.

The EKF maintains two state estimates,  $\hat{\mathbf{X}}(t|t)$  of state  $\mathbf{X}(t)$  and  $\hat{\mathbf{X}}(t+1|t)$  of state  $\mathbf{X}(t+1)$ , as well as two error covariance matrix of above estimates,  $P(t|t)$  and  $P(t+1|t)$ , given observations  $\mathbf{Z}(t), \mathbf{Z}(t-1), \dots, \mathbf{Z}(1)$ . As Fig. 1-4 shows, the EKF is an iterative algorithm, where each iteration involves the four steps. Knowing  $\hat{\mathbf{X}}(t|t)$  and  $P(t|t)$  currently, we have a new observation  $\mathbf{Z}(t+1)$  on the next round and the iteration performs the following steps.

- (1) State prediction:  $\hat{\mathbf{X}}(t+1|t) = f(\hat{\mathbf{X}}(t|t))$
- (2) Observation prediction:  $\hat{\mathbf{Z}}(t+1|t) = h(\hat{\mathbf{X}}(t+1|t))$
- (3) Observation residual:  $\hat{V}(t+1) = \mathbf{Z}(t+1) - \hat{\mathbf{Z}}(t+1|t)$
- (4) State update:  $\hat{\mathbf{X}}(t+1|t+1) = \hat{\mathbf{X}}(t+1|t) + G(t)\hat{V}(t+1)$

where  $G(t)$  is called Kalman Gain (defined later). The basic idea of state update is to use estimated covariance and residual covariance to determine how much we trust on state estimation and how much we trust on the observation residual.

The Kalman Gain and error covariance can be estimated as follows.

- (1) State prediction covariance:  
 $P(t+1|t) = F(t)P(t|t)F(t)^T + Q(t)$
- (2) Observation prediction covariance:  
 $S(t+1) = H(t+1)P(t+1|t)H(t+1)^T + R(t+1)$
- (3) Update Kalman Gain:  
 $G(t+1) = P(t+1|t)H(t+1)^T S(t+1)^{-1}$
- (4) Update state covariance:  
 $P(t+1|t+1) = P(t+1|t) - G(t+1)S(t+1)G(t+1)^T$

where the state transition and observation matrices,  $F(t)$  and  $H(t)$ , are defined as the following Jacobians:

$$F(t) = \left. \frac{\partial f}{\partial x} \right|_{\hat{\mathbf{X}}(t|t)} \quad \text{and} \quad H(t+1) = \left. \frac{\partial h}{\partial x} \right|_{\hat{\mathbf{X}}(t+1|t)}$$

It should be noted that the state and measurement models, *i.e.* the function  $f(\cdot)$  and  $h(\cdot)$  in Eqn. 10 will vary from one problem to another, and they have to be derived separately for each individual problem. Next two parts discuss the details of the problem how to trace the ball states.

### B. Modeling Stochastic Process

**Motion state:** Following the most popular model for kinematic state, we adopt the Continuous White Noise Acceleration (CWNA) model [19] to build the control ball's *motion state*. CWNA assumes the object moves at a constant velocity in the absence of a continuous white acceleration noise, viewing the inputs (*i.e.* human control) as random variables. Combining

Eqn. 4 and Eqn. 5, we have four motion equations:

$$\mathcal{O}(t+1) = \mathcal{O}(t) + \Delta\mathcal{O}(t) \quad (12)$$

$$\mathcal{R}(t+1) = \mathcal{R}(t) \cdot \Delta\mathcal{R}(t) \quad (13)$$

$$\Delta\mathcal{O}(t+1) = \Delta\mathcal{O}(t) \quad (14)$$

$$\Delta\mathcal{R}(t+1) = \Delta\mathcal{R}(t) \quad (15)$$

The four equations above directly determine the ball's state. Both  $\mathcal{R}(t)$  and  $\Delta\mathcal{R}(t)$  are matrices with nine elements, which cannot be in a state vector. However, the  $\mathcal{R}(t)$  are governed by  $\phi_x(t)$ ,  $\phi_y(t)$  and  $\phi_z(t)$  in roll-pitch-yaw expression. Likewise,  $\Delta\mathcal{R}$  is governed by  $\Delta\phi_x(t)$ ,  $\Delta\phi_y(t)$  and  $\Delta\phi_z(t)$ . Thus, we define ball's motion state in form of Eqn. 10 as follows:

$$\mathbf{X}(t) = [\mathcal{O}(t) \ \Phi(t) \ \Delta\mathcal{O}(t) \ \Delta\Phi(t)]^T \quad (16)$$

where  $\Delta\mathcal{O} = (\Delta x_0 \ \Delta y_0 \ \Delta z_0)$  and  $\Delta\Phi = (\Delta\phi_x \ \Delta\phi_y \ \Delta\phi_z)$ . Apparently, the ball's state is a part of its motion state.

**Transition function:** In order to find the transition function  $f(\cdot)$  in Eqn. 10, we need to express  $\Phi(t+1)$  in terms of  $\Phi(t)$  and  $\Delta\Phi(t)$ . Combining Eqn. 3 and Eqn. 13, we have

$$\text{RPY}(\Phi(t+1)) = \text{RPY}(\Phi(t)) \cdot \text{RPY}(\Delta\Phi(t)) \quad (17)$$

Let  $\text{RPY}^{-1}$  be inverse operator of  $\text{RPY}$ , which means that  $\text{RPY}^{-1}$  takes a rotation matrix as the operand and returns its roll-pitch-yaw angles, whereas  $\text{RPY}$  takes roll-pitch-yaw angles as operands and returns the corresponding rotation matrix. Applying  $\text{RPY}^{-1}$  on both sides of Eqn. 17 yields

$$\Phi(t+1) = \text{RPY}^{-1}(\text{RPY}(\Phi(t)) \cdot \text{RPY}(\Delta\Phi(t))) \quad (18)$$

In summary, the transition function  $f(\mathbf{X}(t))$  can be written as the following piecewise function:

$$f(\mathbf{X}(t)) = \begin{cases} \mathcal{O}(t+1) = \mathcal{O}(t) + \Delta\mathcal{O}(t) \\ \Phi(t+1) = \text{RPY}^{-1}(\text{RPY}(\Phi(t)) \cdot \text{RPY}(\Delta\Phi(t))) \\ \Delta\mathcal{O}(t+1) = \Delta\mathcal{O}(t) \\ \Delta\Phi(t+1) = \Delta\Phi(t) \end{cases} \quad (19)$$

The last thing is the explicit expression of  $\text{RPY}^{-1}$ . From Eqn. 3, since  $\phi_y(t)$  is in  $[-\pi/2, \pi/2]$ , the sign of  $\mathcal{R}[1, 1](t)$  is same with  $\cos(\phi_z(t))$  and the sign of  $\mathcal{R}[2, 1](t)$  is same with  $\sin(\phi_z(t))$ . Then we have  $\phi_z(t) = \text{atan2}(\mathcal{R}[2, 1](t), \mathcal{R}[1, 1](t))$ . Similarly, we have  $\phi_x(t) = \text{atan2}(\mathcal{R}[3, 2](t), \mathcal{R}[3, 3](t))$ . As  $\phi_y(t)$  is from  $-\pi/2$  to  $\pi/2$ , we can infer it with an atan function easily.

$$\text{RPY}^{-1}(\mathcal{R}(t)) = \begin{cases} \phi_z(t) = \text{atan2}(\mathcal{R}[2, 1](t), \mathcal{R}[1, 1](t)) \\ \phi_y(t) = \text{atan}(-\mathcal{R}[3, 1](t), (\mathcal{R}^2[2, 1](t) + \mathcal{R}^2[1, 1](t))^{1/2}) \\ \phi_x(t) = \text{atan2}(\mathcal{R}[3, 2](t), \mathcal{R}[3, 3](t)) \end{cases} \quad (20)$$

where  $\text{atan}(\cdot)$  and  $\text{atan2}(\cdot)$  are two types of arc-tangent functions. The  $\mathcal{R}[i, j]$  means the term of matrix  $\mathcal{R}$  (referring to Eqn. 3) at the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column.

### C. Modeling Observations

**Observation vector:** Upon receiving a new round of readings, we calculate all tags' displacements happened between last and current round by using Eqn. 9. There is a relationship between these displacements and our unknown

motion state. The observation model, which is a description of this relationship, is derived in sequel. Let  $\Delta d_{m,n}$  be the displacement of tag  $T_n$  with regards to antenna  $A_n$ . We have the observation vector corresponding to Eqn. 11 as follows

$$\mathbf{Z}(t) = [\Delta d_{1,1}(t) \ \Delta d_{1,2}(t) \ \cdots \ \Delta d_{M,N}(t)]^T \quad (21)$$

The cardinality of observation usually equals  $M \times N$ . We adopt the tags' displacements for observations instead of their absolute positions because the displacement  $\Delta d$  is derived from the phase difference  $\Delta\theta$ , which brings about two obvious benefits. First, the phase difference can eliminate the device diversity. Second, the similar impacts taken by the Doppler effect during a short interval are subtracted by phase difference.

**Observation function:** At time  $t$ , we have the state estimate

$$\hat{\mathbf{X}}(t|t) = [\hat{\mathcal{O}}(t|t) \ \hat{\Phi}(t|t) \ \hat{\Delta\mathcal{O}}(t|t) \ \hat{\Delta\Phi}(t|t)]^T$$

Since we know the transition function,  $\hat{\mathbf{X}}(t+1|t) = f(\hat{\mathbf{X}}(t|t))$ , which is expressed in Eqn. 19, we can predicate the ball's motion state in the next time  $t+1$  using Eqn. 19. Meanwhile, we have the predicted rotation matrix

$$\hat{\mathcal{R}}(t+1|t) = \text{RPY}(\hat{\Phi}(t+1|t)) \quad (22)$$

Therefore, the predicted locations in world coordinate system for tag  $T_n$  at time  $t$  and  $t+1$ , base on Eqn. 2, are given by

$$\begin{aligned} \hat{T}_n^w(t|t) &= \hat{\mathcal{R}}(t|t)T_n^b + \hat{\mathcal{O}}(t|t) \\ \hat{T}_n^w(t+1|t) &= \hat{\mathcal{R}}(t+1|t)T_n^b + \hat{\mathcal{O}}(t+1|t) \end{aligned} \quad (23)$$

where  $T_n^b$  is the actual coordinate in coordinate system  $\mathbf{B}$ . Then the tag displacement can be obtained

$$\Delta \hat{d}_{m,n} = |A_m^w \hat{T}_n^w(t+1|t)| - |A_n^w \hat{T}_n^w(t|t)| \quad (24)$$

where  $A_m^w$  is the  $m^{\text{th}}$  antenna's location. Then the observation function  $h(\cdot)$  corresponding to Eqn. 11 is given by

$$h(\hat{\mathbf{X}}(t+1|t)) = [\Delta \hat{d}_{1,1}, \cdots, \Delta \hat{d}_{M,N}]^T \quad (25)$$

Through the observation function, we can infer a predicated observation  $\hat{\mathbf{Z}}(t+1|t) = h(\hat{\mathbf{X}}(t+1|t))$  based on the predicated state  $\hat{\mathbf{X}}$ . By comparing the measured value  $\mathbf{Z}$  (shown in Eqn. 21) and the predicated  $\hat{\mathbf{Z}}$  (Eqn. 25), the observation residual is calculated, which will be used for state update in the last step of EKF. If the phase value measured by the  $m^{\text{th}}$  antenna for the  $n^{\text{th}}$  tag is missed, we delete the corresponding items in  $\mathbf{Z}(t)$ ,  $\hat{\mathbf{Z}}(t+1|t)$  and  $R(t)$  and the EKF still works.

## VI. SMOOTHING BALL STATE

The EKF only computes the estimates conditioned on previous and current measurements instead of the measurements over a time window  $w$ . Tagball employs the Rauch-Tung-Striebel (RTS) smoother to smooth the changes. After receiving a set of observations  $\mathbf{Z}(t), \mathbf{Z}(t-1), \cdots, \mathbf{Z}(t-w)$ , we are going to compute the estimate of the whole trajectory with minimum mean square error, which is conditioned on all the measurements

$$\hat{\mathbf{X}}^s(t) = \text{E}[\mathbf{X}(t)|\mathbf{Z}(t), \mathbf{Z}(t-1), \cdots, \mathbf{Z}(t-w)] \quad (26)$$

where  $w$  is the smoothing window size. The estimate can be computed with the RTSI smoother. The smoothed state estimate

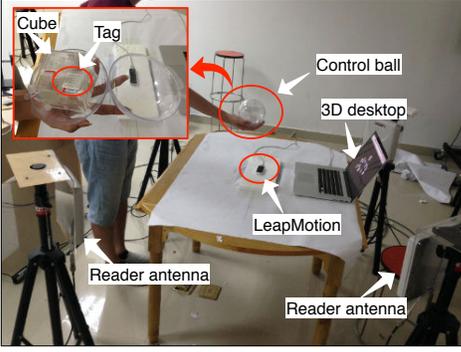


Fig. 2: System implementation

and its covariance can be computed by using the following backward recursion for these states during the time window.

$$\begin{aligned}
 \hat{\mathbf{X}}(t|t-1) &= F(t-1)\hat{\mathbf{X}}(t-1|t-1) \\
 P(t|t-1) &= F(t-1)P(t-1)F(t-1)^T + W(t-1) \\
 G(t-1) &= P(t-1|t-1)F(t-1)^T[P(t|t-1)]^{-1} \\
 \hat{\mathbf{X}}^s(t-1) &= \hat{\mathbf{X}}(t-1|t-1) + G(t-1)[\hat{\mathbf{X}}^s(t) - \hat{\mathbf{X}}(t|t-1)] \\
 \mathbf{P}^s(t-1) &= \mathbf{P}(t-1) + G(t-1)[\mathbf{P}^s(t) - \mathbf{P}(t|t-1)]G^T(t-1)
 \end{aligned} \tag{27}$$

the recursion starts from the filter results at the current time step  $\hat{\mathbf{X}}^s(t) = \hat{\mathbf{X}}(t|t)$ ,  $\mathbf{P}^s(t) = \mathbf{P}(t|t)$ . When the data set is limited and the assumed initial position differs significantly from the ground truth, it is possible that the basis estimation does not converge close enough to the true bias during one run of the filter and the smoother. Then it is possible to iteratively run the filter and smoother back and forth until the change in the bias between consecutive iterations is below a threshold.

## VII. IMPLEMENTATION AND EVALUATION

This section presents the implementation of our prototype and the evaluation with various actions and parameters.

### A. Implementation and Setup

We build a prototype of Tagball using COTS reader and tags with standard communication protocol, as shown in Fig. 2.

- **Control ball:** It is inconvenient to attach tags on a spherical surface. While implementing, we attach the tags on a cube whose edge length is  $6.5\text{cm}$  and embed the cube inside a plastic ball (its radius equals  $6\text{cm}$ ), as illustrated in Fig. 2. Each side of the cube has two passive tags modeled Square from Alien company. Totally, 12 tags are attached. In addition, we add 8 paddings between the ball and cube to stabilize the tags' geometric relationships. Before evaluation, we perform a set of calibration experiments to eliminate the device diversity [6] for each pair of antenna and tags.

- **RF receiver:** We employ an Impinj Speedway modelled R420 reader, which supports four directional antennas at most, being compatible with EPC Gen2 standard. The reader works in the  $920 \sim 926$  MHz band with frequency hopping. The reader communicates a host through the wireless network (TCP/IP). An average read-out speed of the reader is around 400 times per second (100 read-outs per second from each reader antenna). The four antennas with circular polarization, manufactured by Yeon technology are connected to reader,

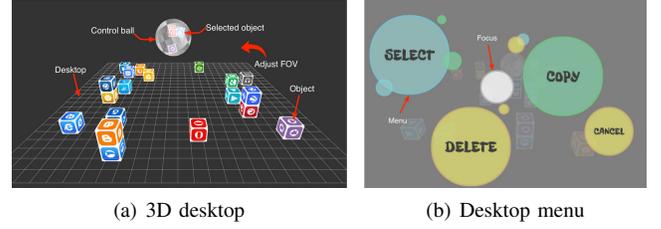


Fig. 3: Three dimensional desktop

providing  $\geq 8\text{dBic}$  gain in two directions. The size of each antenna equals  $225\text{mm} \times 225\text{mm} \times 40\text{mm}$ . The four antennas are deployed at the corners of a square with a side length of  $1.2\text{m}$  and kept  $2\text{m}$  above the floor.

- **Software:** We adopt Low Level Reader Protocol (LLRP) to communicate with the reader. Impinj reader extends this protocol to support the phase report. We adjust the configuration of reader to immediately report reading whenever tag is detected. The client code is coded in C# programming language. Based on Tagball system, we develop an application called *3D desktop*. The screen shot of 3D desktop is shown in Fig. 3. A user can select, copy, delete, and rearrange the objects on the desktop. He also can adjust the field of view through the rotation operations. This application has two implementation parts. One part is run at the client and connect to the Tagball system and the other part is implemented using Tree.js (a JavaScript 3D Library which makes WebGL). Two parts communicate to each other with Webservice.

- **Ground truth:** The LeapMotion [20] is a recent novel interaction device used to capture the fingers' motion, which is able to achieve an accuracy of  $0.001\text{mm}$ . Thus, we use LeapMotion to capture the ground truth. We place the device under the control ball and let volunteer keep palms facing up.

### B. Translation Accuracy

We firstly present the translation accuracy with other 7 localization or tracking methods. We repeat the experiments over 100 measurements. To focus on how much the reconstructed translation shape deviates from the actual shape, we let the control ball move along a piecewise linear curve, and calculate the segment-by-segment length difference between the shifted reconstructed translation and the ground truth. Note that each segment of the trajectory has a random length and direction. Fig. 4 shows the translation errors for 8 methods.

- **RSS:** The difference between the RSSs of a pair of tags is used as an indicator for their spatial distance in past work [21]. The RSS scheme has an error distance of  $60\text{cm}$ , suffering from the high variation in behavior across tags, like the antenna gain and tag's orientation.

- **PinIt:** In PinIt [4], a SAR is created through a mobile antenna to extract the multipath profile for each tag. Its intuition is that the tags locating in similar environment have similar multipath profiles. PinIt achieves a mean error distance of  $20\text{cm}$  around.

- **BackPos:** BackPos [22] introduces the technique of hyperbolic positioning into RFID localization without the need of reference tags. It obtains a mean error distance of  $40\text{cm}$  with

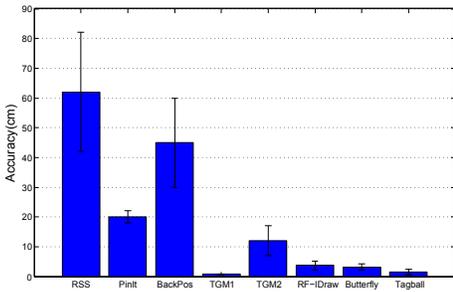


Fig. 4: Accuracy comparisons

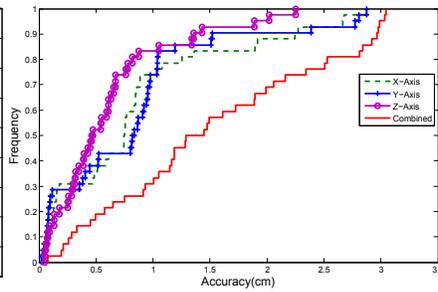


Fig. 5: Translation accuracy

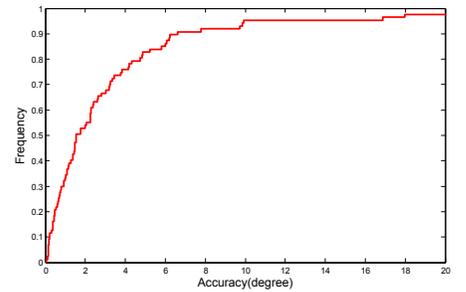


Fig. 6: Rotation accuracy

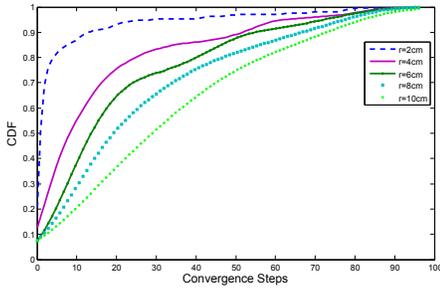


Fig. 7: Convergence rate

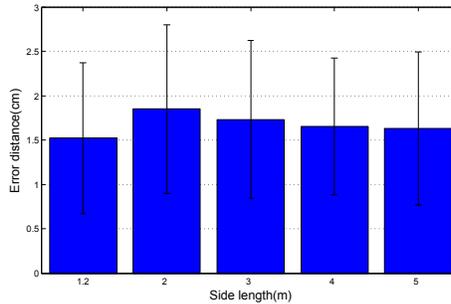


Fig. 8: Impact of distance

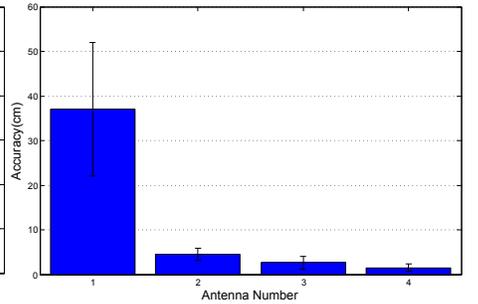


Fig. 9: Impact of antenna number

a standard deviation of  $20cm$ . It sacrifices the area of feasible region to obtain the feature of being anchor-free.

- **Tagoram:** Tagoram [6] discusses two cases. TMG1 means the controllable case in which the track function is known in advance. Using the technique of differential augmented hologram, Tagoram is able to achieve a median error distance of  $1.4cm$ . TMG2 is the uncontrollable case where the track function is unknown, in which it obtains a median error distance of  $12cm$ .

- **RF-IDraw:** Like BackPos, RF-IDraw also employs the technique of hyperbolic positioning to localize the tag's trajectories, but it focuses on resolving the problem of multi-resolution. RF-IDraw can be used for users to interact with computing devices by gesturing or writing their commands in the air. It can track the trajectory shape of the user's writing with a median accuracy of  $3.7cm$ .

- **Butterfly:** [18] presents RFID-based insect tracking system that equips a butterfly with a passive tag. It also utilizes the EKF to resolve the ambiguity of the phase measurements and obtain location accuracy of  $3.2cm$ .

- **Tagball:** Fig. 5 shows the detailed CDF of translation accuracy for Tagball. The median errors are  $0.7cm$ ,  $0.7cm$ ,  $0.5cm$  and  $1.5cm$  along  $x$ -axis,  $y$ -axis,  $z$ -axis and combined dimension. Its 90th percentile errors are  $2.2cm$ ,  $2.0cm$ ,  $1.4cm$  and  $3cm$ , respectively. There is little difference among different dimensions. Hence, TagBall outperforms the translation accuracy over RSS, PinIt, BackPos, TMG2, RF-IDraw and Butterfly by over  $41\times$ ,  $13\times$ ,  $30\times$ ,  $8\times$ ,  $2.4\times$  and  $2.1\times$ .

**Summary:** Despite the high precision of TMG1, it cannot be used in our scenario because the user behavior is totally unpredictable. Except TMG1, Tagball achieves the best accuracy among these methods. RF-IDraw has the idea similar to ours. But RF-IDraw considers the tag's trajectory as a series of isolated points instead of a continuous process, without

considering of motion contexts. the major difference between Butterfly and Tagball is that the latter takes the behaviors of group tags into consideration and provides the additional rotation inspection, while the former only focuses on the behavior of individual tags.

### C. Rotation Accuracy

Second, we investigate the rotation accuracy. In order to determine the rotation of an object, at least two tags need to be attached and read on the object. We place the control ball at a fixed position and rotate it with random angles. We calculate the rotation difference between the inferred and the ground truth. For baseline, we compare the results with the RF-Compass.

- **RF-Compass:** RF-Compass [5] is an application of PinIt [4] in robot object manipulation, which is designed to navigate a robot equipped with RFIDs toward the object. Through the spatial partitioning, it can identify the orientation of the object so that the robot may pick the object up. RF-Compass can achieve a median of rotation error of  $3.3^\circ$ .

- **Tagball:** Fig. 6 shows the CDF of rotation accuracy identified by Tagball. It has a median rotation error of about  $1.8^\circ$  and 90<sup>th</sup> percentile error is  $6.3^\circ$ , outperforming the accuracy over RF-Compass by  $1.8\times$ .

Being different from the RF-Compass which locates each tag firstly and then infer the object's orientation, Tagball considers the orientation (or rotation) as one dimension of the ball state without knowing each tag's location.

### D. Impacts of Parameters

Last, we consider other factors that may take impacts on the accuracy.

- **Impact of initial state:** The estimated initial ball state has influence on the convergence rate of EKF. We place control ball around the ground truth with various distance  $r$ , and

then observe the influence of the estimated error. Fig. 7 plots the CDF of converge steps needed by EKF to obtain a high precision. We observe that the EKF can fast converge when the error is very small. The accuracy and run-time of initial ball state estimating is relevant to the scope and granularity of searched state space. To reduce the initialization time, we suggest the user set the ball in a specified orientation state at first. As a coarse initial state estimate is sufficient, the orientation state need not be very accurate so that the user need not rotate the ball very carefully. And we think the initial step does not cause inconvenience.

• **Impact of distance:** We deploy the four antennas at the four corners of a square respectively. The control ball performs actions inside the square. In order to check the impacts of distance between the reader and control ball, we adjust the side length of the square with  $1.2m$ ,  $2m$ ,  $3m$ ,  $4m$  and  $5m$  respectively. Fig. 8 shows the accuracy with varying distance. There is little difference among these distances, indicating that the tracing accuracy is irrelevant to the distance.

• **Impact of antenna number:** The advantage of modelling the control ball as a whole and viewing the phase measurements as observations, is that our system is not restrained by the number of antennas. We change the number from 1 to 4 and see its impact on the accuracy. The results are shown in Fig. 9, the accuracy extremely improves as the number increases. Especially, the median accuracy is  $6cm$  even when we use two antennas. This feature benefits from the statistical characteristics of EKF.

### VIII. CONCLUSION

This paper presents an RFID-based interaction device for human-computer device in 3D space. Tagball bridges the human-computer barrier, facilitating natural human-computer interaction in 3D world. Our key innovations are studies on motion behaviors of a group of tags by using Extended Kalman Filter, and the implementation based on purely COTS RFID products. In the future, we will further improve the speed by reducing the collision among reader and tags [23]–[25].

### ACKNOWLEDGEMENT

This study is supported in part by the NSF China Major Program No. 61190110, NSF China Key Program No. 61432015, NSF China Projects No. 61472211 and 61428203. The research of Li is partially supported by NSF ECCS-1247944, NSF CMMI 1436786, National Natural Science Foundation of China under Grant No. 61170216, No. 61228202.

### REFERENCES

- [1] J. Han, H. Ding, C. Qian, D. Ma, W. Xi, Z. Wang, Z. Jiang, and L. Shangguan, "Cbid: A customer behavior identification system using passive tags," in *Proc. of IEEE ICNP*, 2014.
- [2] G. Cohn, D. Morris, S. Patel, and D. Tan, "Humantenna: using the body as an antenna for real-time whole-body interaction," in *Proc. of ACM SIGCHI*, 2012.
- [3] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. of ACM MobiCom*, 2013.
- [4] J. Wang and D. Katabi, "Dude, where's my card?: Rfid positioning that works with multipath and non-line of sight," in *Proc. of ACM SIGCOMM*, 2013.
- [5] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus, "Rf-compass: robot object manipulation using rfids," in *Proc. of ACM MobiCom*, 2013.
- [6] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile tag to high precision using cots devices," in *Proc. of ACM MobiCom*, 2014.
- [7] L. Shangguan, Z. Li, Z. Yang, M. Li, and Y. Liu, "Otrack: Order tracking for luggage in mobile rfid systems," in *Proc. of IEEE INFOCOM*, 2013.
- [8] C.-S. Fahn and H. Sun, "Development of a data glove with reducing sensors based on magnetic induction," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 2, pp. 585–594, 2005.
- [9] K. N. Tarchanidis and J. N. Lygouras, "Data glove with a force sensor," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 3, pp. 984–989, 2003.
- [10] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proc. of ACM MobiSys*, 2011.
- [11] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tappprints: your finger taps have fingerprints," in *Proc. of ACM MobiSys*, 2012.
- [12] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *Proc. of USENIX NSDI*, 2014.
- [13] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proc. of ACM SIGCHI*, 2012.
- [14] J. Wang, D. Vasisht, and D. Katabi, "Rf-idraw: Virtual touch screen in the air using rf signals," in *Proc. of ACM SIGCOMM*, 2014.
- [15] S. Azzouzi, M. Cremer, U. Dettmar, R. Kronberger, and T. Knie, "New measurement results for the localization of uhf rfid transponders using an angle of arrival (aoa) approach," in *Proc. of IEEE RFID*, 2011.
- [16] C. Hekimian-Williams, B. Grant, X. Liu, Z. Zhang, and P. Kumar, "Accurate localization of rfid tags using phase difference," in *Proc. of IEEE RFID*, 2010.
- [17] J. Wu, R. Rink, T. M. Caelli, and V. Gourishankar, "Recovery of the 3-d location and motion of a rigid object through camera image (an extended kalman filter approach)," *International Journal of Computer Vision*, vol. 2, no. 4, pp. 373–394, 1989.
- [18] S. Sarkka, V. V. Viikari, M. Huusko, and K. Jaakkola, "Phase-based uhf rfid tracking with nonlinear kalman filtering and smoothing," *Sensors Journal, IEEE*, vol. 12, no. 5, pp. 904–910, 2012.
- [19] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [20] "Leap Motion," <https://www.leapmotion.com>.
- [21] L. Ni, Y. Liu, Y. Lau, and A. Patil, "Landmarc: Indoor location sensing using active rfid," *Wireless networks*, 2004.
- [22] T. Liu, L. Yang, Q. Lin, Y. Guo, and Y. Liu, "Anchor-free backscatter positioning for rfid tags with high accuracy," in *Proc. of IEEE INFOCOM*, 2014.
- [23] J. Lim, S. Kim, H. Oh, and D. Kim, "A designated query protocol for serverless mobile rfid systems with reader and tag privacy," *Tsinghua Science and Technology*, vol. 17, no. 5, pp. 521–536, 2012.
- [24] C. Qian, Y. Liu, R. H. Ngan, and L. M. Ni, "Asap: Scalable collision arbitration for large rfid systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 7, pp. 1277–1288, 2013.
- [25] Y. Zheng and M. Li, "Fast tag searching protocol for large-scale rfid systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 3, pp. 924–934, 2013.