Catching a Real Ball in Virtual Reality

Matthew K.X.J. Pan*
Disney Research

Günter Niemeyer†
Disney Research

Abstract

We present a system enabling users to accurately catch a real ball while immersed in a virtual reality environment. We examine three visualizations: rendering a matching virtual ball, the predicted trajectory of the ball, and a target catching point lying on the predicted trajectory. In our demonstration system, we track the projectile motion of a ball as it is being tossed between users. Using Unscented Kalman Filtering, we generate predictive estimates of the ball's motion as it approaches the catcher. The predictive assistance visualizations effectively increases the user's senses but can also alter the user's strategy in catching.

Keywords: Tracking, virtual reality, motion prediction, physical interaction, mixed reality, catching.

Index Terms: Human-centered computing → Human computer interaction (HCI) → Interaction paradigms → Virtual reality; Human-centered computing → Human computer interaction (HCI) → Interaction paradigms → Mixed / augmented reality.

1 Introduction

In this work, we explore haptic sensations in virtual reality (VR) promoting the idea of users interacting with *dynamic* physical objects. To demonstrate this concept, we consider the catching of a real ball while immersed in virtual reality. From a psychomotor perspective, catching a thrown ball by hand is not an easy task but demands many coordinated skills that are learned from childhood. It takes acute visual sensing, prediction, temporal planning and refined motor control to catch. Performing such a task in VR offers an even greater challenge as other external factors such as system latency, rendering of depth cues, frame rate, tracking precision, and registration affect performance. We perceive this problem as a first step towards more complicated dynamic object interactions which can be used to further immerse users in virtual environments.

We consider three visualizations intended to aid users in catching real balls while immersed in VR:

- **Virtual Ball**: a virtual ball is rendered which tracks the real ball
- **Trajectory**: the predicted trajectory of the real ball is displayed using a line (Figure 1A), and
- **Target**: a target catching location lying on the predicted trajectory of the real ball is displayed. As under-hand catches occur at an approximately constant height, we define the target as the ball location when it drops to this height. The target object also indicates direction in which the ball will be arriving to allow users to correctly orient their hands (Figure 1B).

These visualizations may be applied separately or in combination with each other.

*matt.pan@disneyresearch.com
†gunter@disneyresearch.com

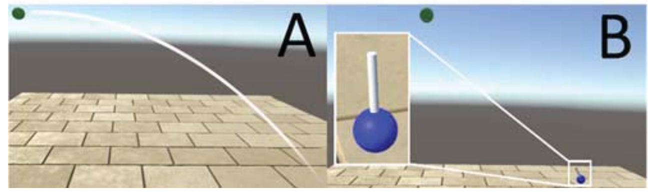

Figure 1. Forms of VR catching visualization: (A) Trajectory, where the predicted trajectory of the real ball is rendered. (B) Target, where a blue sphere indicates the target position and a white stick indicates the direction of arrival. In both panes, the virtual ball is shown.

2 System

To provide these visualizations, we use an OptiTrack Flex 13 motion capture system to track the motion of a ball as well as the catcher's hands and head at 120 fps and mean lag of 8.33ms. The system predicts the future trajectory of the ball as it undergoes projectile motion, and renders a virtual scene to display the ball's location as well as any assistive cues. Our virtual scene is rendered via a head-mounted display (Oculus CV1) using the Unity 3D game engine on a Windows 10 x64 system (Intel Xeon E5-2680 2.5 GHz processor, 32 GB RAM, and NVidia GeForce GTX 970 graphics). The environment was deliberately kept minimalistic to maximize frame rate (140-150 fps) and minimize latency. As such, our scene only contains a textured floor, a virtual ball, 5 cm x 5 cm square paddles representing the user's hands, and Unity default lighting.

3 Object Tracking and Prediction

We employ a discrete time Unscented Kalman Filter (UKF) to filter position and velocity data of the ball generated by the motion capture system [1]. The nonlinear UKF is able to integrate a quadratic aerodynamic drag

$$\vec{F}_D = -D\ \|\vec{v}\|\vec{v} \tag{1}$$

into the ball's flight dynamics model, where D is a drag constant and $\vec{v}$ is the velocity. We define the drag constant as

$$D = \frac{\rho C A}{2} \tag{2}$$

where ρ is the density of air, C is the drag coefficient (approximated as 0.5 for a sphere) and A is the cross sectional area of the ball. The full model thus computes the current acceleration

$$m\vec{a}[k] = -m\,\vec{g} - D\|\vec{v}[k]\|\ \vec{v}[k] \tag{3}$$

where m is the ball mass and $\vec{g}$ is the gravitational acceleration vector. It also updates the velocity

$$\vec{v}[k+1] = \vec{v}[k] + \vec{a}[\mathrm{k}]\Delta t \tag{4}$$

and position

$$\vec{p}[k+1] = \vec{p}[k] + \vec{v}[k]\Delta t + \tfrac{1}{2}\vec{a}[k]\Delta t^2 \tag{5}$$

over the time step Δt. For the UKF we use a process noise of 0.01 and measurement noise of 0.005.

At each time step, we predict the real ball's entire future trajectory to generate the trajectory and target visualizations, integrating the dynamic model (3-5) forward in time from the

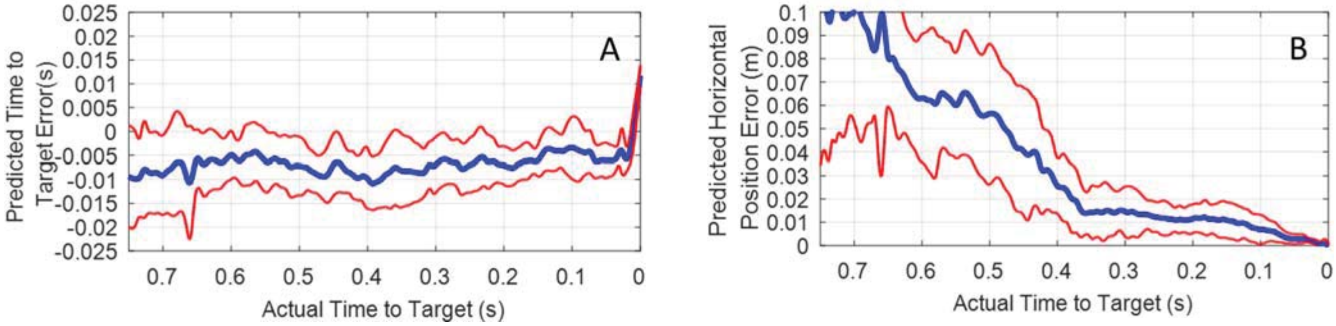

Figure 2. Predicted time to target errors vs actual time to target (where the target was the actual location of the ball when it was 10 cm above the ground) (A) and Distance between the predicted and actual target positions vs. time to target (B). The thick blue lines represents the mean errors across all trials while the thin red lines represent one standard deviation above and below the mean.

current position and velocity estimates. We terminate this integration and hence the future trajectory prediction when a target criterion has been reached, for example, when the ball drops to a specified height. Using this trajectory, we are thus able to predict the target position and estimated time that it will take for the ball to arrive at the target. These predictions evolve and should naturally refine as time advances.

3.1 Performance

To evaluate prediction accuracy, we tossed a ball using an underarm throw and continually estimated the horizontal target position and time to target. Without an explicit catch, the target is defined as the point where the ball drops to a height of 10cm above the floor. These predictions were recorded for each update and later compared against the actual time and position. In Figure 2 we show the results of 11 tosses, presenting an average prediction error and a standard deviation width. The mean duration of all ball tosses was 1.087s in duration, but we graph only the final 0.75s before the target time, so that all data is collected in flight.

Time to target estimates, as seen in Figure 2A, converge fairly early and by 0.7s to target are within 5-10ms. We believe the residual 5ms error as well as the final sharp change in error are mostly an artifact of the 8ms discrete sampling time. In Figure 2B we show the 2D Euclidian distance between the predicted target position and the position of the ball when it was at 10cm above the floor as a function of time to target. We see that by 0.7s to target the prediction falls within a 10cm radius of the target. By 0.35s to target it plateaus below 2cm.

From research in psychomotor behavior [2], we know that humans need trajectory information at least 200ms before a catch to be able to react. We thus conclude that the information gathered by the prediction, if correctly communicated to users, will allow them to catch.

4 Preliminary Results

We tested the effectiveness of catching using our system over the course of 140 tosses (20 tosses for each combination of the three visualizations excluding the condition where no visualization is presented) in a pilot study. In total 132 balls were caught, underlining the overall success of the user's ability to catch in VR.

20 of these tosses were made with only the virtual ball which most closely matches how balls are caught in the physical world. In this condition, 95% of balls were caught, indicating that our system allows users to catch reliably. Video and screen capture footage indicate that during the catch, the user visually focuses on the trajectory of the ball and does not keep their hands within viewing range until just before the catch. From this evidence, it can be inferred that proprioception is used to position the hands using visual and depth cues of the ball.

Catching with the other visualizations did not seem to affect catching behaviors, except in the cases where the virtual ball was not rendered: the removal of the virtual ball from the VR scene seems to allow the catcher's hands to reach the catch location much earlier prior to catching. The most apparent explanation for this phenomenon lies with the observation that the user is forced to alter catching strategy: the catcher has to rely on the target point/trajectory and so the motor task has changed from a catching task, which had required higher brain functions to estimate the trajectory, to a simpler, visually-guided pointing task requiring no estimation at all.

5 Discussion and Conclusions

In this work, we have presented a proof of concept system which enables users to catch a tossed physical ball while in VR. It appears that while small latencies exist in our system and objects such as the user's hands are rendered abstractly, our system allows users to be quite adept at catching balls while in VR. Thus, combining virtual and physical dynamic interactions to enrich virtual reality experiences is feasible. Additionally, we have implemented low-error prediction of the ball's trajectory using a non-linear dynamic model. In turn, these predictions inform tools to assist a user in catching ball. We have demonstrated several methods in an attempt to enhance the catching task within VR with modest success. In one instance, we have discovered that when a predicted target location is displayed without rendering the ball, users are forced to switch catching behaviors: rather than having users to predict the trajectory of the ball to make the catch, we have reduced cognitive burden and transformed the task into a simpler pointing task. Thus, this result shows that this task model for catching appears to be more efficient from a psychomotor perspective.

We believe this work provides valuable insight which informs how interactions with dynamic objects can be achieved while users are immersed in VR. As a result of these preliminary findings, we have discovered many more avenues for future work in dynamic object interactions in VR.

References

[1] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[2] J. López-Moliner, E. Brenner, S. Louw, and J. B. J. Smeets, "Catching a gently thrown ball," *Exp. Brain Res.*, vol. 206, no. 4, pp. 409–417, Oct. 2010.

[3] A. Murata and H. Iwase, "Extending Fitts' law to a three-dimensional pointing task," *Hum. Mov. Sci.*, vol. 20, no. 6, pp. 791–805, 2001.