Improving Reliability of Virtual Collision Responses: A Cue Integration Technique

Seungwon Do and Byungjoo Lee*
KAIST, Republic of Korea
{monk1,byungjoo.lee}@kaist.ac.kr

ABSTRACT
In virtual reality (VR), a user's virtual avatar can interact with a virtual object by colliding with it. If collision responses do not occur in the direction that the user expects, the user experiences degradation of accuracy and precision in applications such as VR sports games. In determining the response of a virtual collision, existing physics engines have not considered the direction in which the user perceived and estimated the collision. Based on the *cue integration theory*, this study presents a statistical model explaining how users estimate the direction of a virtual collision from their body's orientation and velocity vectors. The accuracy and precision of virtual collisions can be improved by 8.77% and 30.29%, respectively, by setting the virtual collision response in the direction that users perceive.

Author Keywords
Virtual reality; collision response; cue integration theory.

CCS Concepts
•Human-centered computing → Virtual reality;
•Computing methodologies → Perception;

INTRODUCTION
Since the commercialization of head mounted display (HMD) based devices and hand trackers such as the `Leap Motion Controller`, it has become possible to implement VR applications that have a direct interaction between the user's body and virtual objects. To manipulate a virtual object directly through the body, the user's body must first be represented as another virtual object in virtual space, which can be called the user's avatar. The system then detects the collision between the virtual object and the user's avatar (i.e., collision detection) so that the object can be moved in the direction of the collision (i.e., collision response) [47]. This resembles manipulation in the real world, and the user can easily adapt to the interaction.

Virtual collisions between a user's avatar and a virtual object can be detected and simulated by various physics engines. During the collision, the physics engines iteratively return the

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04. . . $15.00
DOI: https://doi.org/10.1145/3313831.3376819

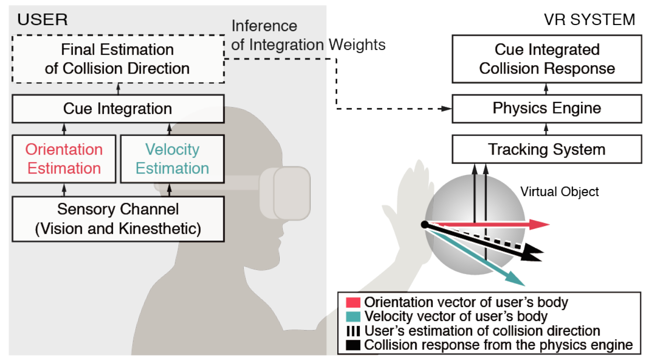

Figure 1. We proposes a novel collision technique that can improve the reliability of virtual collision responses. Based on *cue integration theory*, our technique ensures that the direction of the collision response is the same as the direction of the collision that the user estimates.

magnitude and direction of the virtual impact, which affects the motion of the colliding objects. Based on the laws of physics, the magnitude and direction of the impact are determined by various factors, such as weight, elasticity, and friction [47]. Techniques for simulating the physical collision process for virtual objects have been extensively studied, leading to high simulation accuracy and precision provided by commercial physics engines today [6].

However, from the user's point of view, the virtual collision process is still essentially different from the collision process in the real world. In a physical collision process, the same amount of impact is applied to the user as on the object (i.e., Newton's law of action-reaction), but in a virtual collision, there is no force transmitted from the virtual object to the user. In other words, in the virtual collision process, the user receives different sensory signals than from the physical collisions that have been experienced in the real world [32].

From past experiences of physical collisions, the user's brain has learned a mapping function (i.e., an internal model) between sensory stimuli resulting from the collision and the corresponding consequences of the collision [54]. The user can also use the mapping function to predict the response during the virtual collision. However, the absence of force sensations creates a mismatch between the response of the collision predicted by the user's internal model and the response actually produced by the system. This causes a decrease in the accuracy and precision of the collision behavior that the user plans and performs [33, 18, 2, 44]. In VR sports games such

*Corresponding author

as table tennis, the reduced performance and immersion that players feel may originate from this problem.

There can be three approaches to solving this problem. First, long training time can allow users to get familiar with virtual collisions. The user's internal model itself can be updated through learning about virtual collision, which may take a long time [3], and there is indirect evidence that shows the longer experience in virtual space does not actually lead to improved usability [55, 66] (i.e., simulator sickness). Secondly, we can implement a special haptic device to actually impact the user in virtual collisions. This is the latest approach that studies on VR interaction techniques have taken [24, 29, 37, 68, 63, 9, 28]. If the haptic system is implemented properly, it can provide the user with a situation that closely resembles the actual physical collision, thus bringing the reliability of the collision to a level similar to the physical collision. However, this method also has a disadvantage in that the system is difficult to implement and the usability may be limited due to the bulky hardware.

Thirdly, the approach taken in this study is to figure out how the user's internal model estimates virtual collisions and provides a response accordingly. For example, we can describe what sensory signals are given to the user during the collision and how the user perceives the collision from them, and then provide a response in the exactly way perceived by the user. This method will be useful if implemented properly because it reduces the difference between the user's predicted collision response and the system's actual response, and requires no extra hardware and less training by the user. However, it has been rarely attempted because it was difficult to build a computational model of how users perceived virtual collisions.

Recent advances in cognitive science have enabled a computational approach to human perception. In particular, the *cue integration theory* [16] provides a statistical model that describes the process by which estimates from different sensory channels are integrated into the final estimate. For example, suppose a person perceives the weight of a coffee mug. The person can estimate the weight differently through a visual cue like cup size and the kinesthetic cue felt when actually lifting it. The theory explains that humans perceive the final cup weight through maximum likelihood estimation (MLE) based on the reliability of those two different estimates.

Based on the cue integration theory, this study mathematically describes how users estimate the *direction* of virtual collisions. We assume that in the process of virtual collision, the user can estimate the direction of the virtual collision from the *orientation* and *velocity* of his or her body perceived through the visual and the kinesthetic sensation (see Fig. 1). We can express the reliability of those two directional estimates mathematically and then express the user's final estimation of the collision direction, resulting from the MLE. Finally, our collision technique generates the response of the collision in the user's perceived direction. From this, the accuracy and precision of the virtual collision response can be improved.

The collision technique proposed in this study is implemented and tested for a situation similar to a VR sports game in which a hand, represented by a plate-shaped avatar, hits a ball. The situation where a human-controlled plate collides with a virtual ball is widely used in VR games today. For example, in `Steam`, the most popular online game platform, there are many VR games fitted to this condition such as `Racket:NX`, `Eleven:Table Tennis VR`, `First Person Tennis`, `Magical Squash`, and `Holoball`.

The study is divided into three parts. In the first part, we present a statistical model that describes the user's estimation of the virtual collision direction. The second part describes a calibration user study to determine the free parameters of the proposed model at the population-level. The third part describes a user study comparing the accuracy and precision of our collision technique with a baseline.

RELATED WORK

Simulating Collisions for Virtual Objects

Physically realistic simulation of the collision process between virtual objects is a traditional research topic in computer graphics. The various studies on this study can be subdivided into research on two techniques: *collision detection* and *collision response* [47]. Collision detection techniques literally aim to determine when, where and, how two virtual objects intersect [15]. Collision response techniques determine what forces are applied to two objects due to the detected collision.

Earlier collision detection and collision response studies have focused on simulating physically accurate and realistic collisions. In this case, the important physics related to the collision are elasticity and friction. Elasticity determines the repulsive force between two objects in the direction of the normal vector of the contact surface [47]. Friction determines the force exerted between two objects in the direction of the tangential vector of the contact surface [47]. The physics engines built into 3D development tools such as `Unity` and `Unreal Engine` can consider both the elasticity and friction phenomena in determining the collision response. Although they have different biases between stability and accuracy [14, 27], commercial physics engines are all accurate enough [6] to simulate collisions between virtual objects (not with the user).

Perceptually Adaptive Computer Graphics

In computer graphics, much of the research on the human factors in the design of the physics engine has been carried out mainly on the user's visual perception [52]. Those previous studies explored how graphics simulations are perceived as real to the user [17], and how to lower the level of detail (LOD) of graphics simulations without compromising the realism the user feels in order to speed up the simulation [4, 35]. Humans have some degree of imperfection in visually perceiving and predicting the processes of physical collisions between objects [50]. By utilizing these human characteristics, we can optimize the performance of the system by lowering the LOD of the collision algorithm to the extent that viewers cannot notice the change [51]. The simulation of the virtual collision can also be further improved by utilizing other characteristics of the user's visual perception, such as selective attention [53, 49] and context-dependent perception [58].

A recent study dealt with the issue of user perception in situations where an avatar controlled by a user collides with a virtual object [33]. Boxer, a collision technique proposed in the study, detects collisions at the avatar's minimum speed and improves the precision of collision responses over existing collision techniques. Similar to this study, the Boxer technique considers the user's kinesthetic perception in the process of virtual collisions, but has an additional delay until it detects a collision and is only applicable to special avatar motions where the hand is pulled back immediately after touching the virtual object. The study also did not provide a computational explanation of how the user estimates the collision direction during the virtual collision process.

Augmentation of Direct Manipulation in VR and AR

Various haptic devices have been recently proposed that can be used in VR and AR [24, 29, 37, 68, 63, 9, 28], and they increase the realism and performance of the interaction by applying a force similar to that given to the user in the actual physical collision. However, it is not known whether such force feedback actually improves the user's performance in dynamic applications such as VR sports games. Also, the large volume of the systems and the complex hardware they require are obstacles that make them difficult to put into practical use.

Other studies have proposed heuristic-based object manipulation techniques to overcome the low reliability of virtual collisions [41]. The techniques consider how to map commands such as selection (or grabbing) [2, 65, 1, 5], translation [19, 42, 48], and rotation [42, 48] to the user's body movement, rather than directly improving the response of a collision. Other studies have improved grasping performance by improving how the user's hand is physically represented in virtual space [26, 7, 25]. All these studies do not fundamentally address the perceptual problems of virtual collisions.

Human Perception in Physical Contact

Humans receive information from various sensory channels in the process of colliding their body with physical objects [54], mostly by the pressure and vibration of the contact perceived by the skin mechanoreceptor [30] and the position and velocity of the body perceived by the proprioceptor neuron distributed in muscles and joints [69] (also known as kinesthetic perception). The position of the body can also be perceived by vision, which is complementary to proprioception [62, 21].

The human brain builds an *internal model* [11, 12] of the collision process as it experiences sensory cues and the consequences of collisions. Research on the predictive brain [8] explains that the internal model allows humans to predict in advance the consequences of the collision. When an error between prediction and actual phenomena is perceived, the human brain tries to eliminate the error signal by updating its internal model or by changing its environment through actions [20]. This explains why the lack of haptic sensations in virtual collisions increases the user's cognitive burden and impairs the accuracy and precision of collision planning and execution.

Summary

In summary, due to the lack of repulsive force given to the user in the virtual collision process, the user's internal model frequently sees the error between the predicted and actual phenomena. This can reduce the reliability of the user's movement plan. However, this problem is rarely studied in the field of HCI or computer graphics. By matching the direction of the collision response with the direction of the collision estimated by the user, the novel collision technique proposed in this study is a practical solution to improve the accuracy and precision of the interaction in VR without additional hardware.

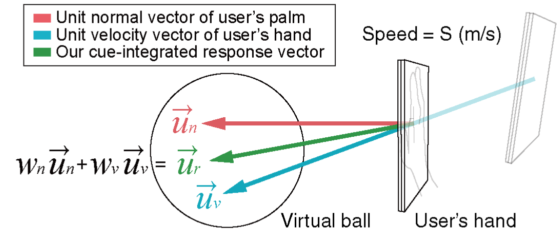

Figure 2. This study deals with a virtual collision between a virtual ball and a user's hand represented by a plate in the virtual space.

MODEL OF COLLISION DIRECTION PERCEPTION

This study proposes a novel collision technique that can generate the response of a collision in the direction perceived by the user. To do this, we first build a statistical model explaining how the user estimates the collision direction and how the reliability of the estimation changes with the collision speed.

Problem Formulation

In this study, we first assume that to simplify the modeling process, the user's body is represented as a *single rigid body* in virtual space and it collides with the virtual object. More specifically, this study assumes that the user strikes an object with his or her palm and that the user's hand is represented as a plate-shaped rigid body in virtual space (see Fig. 2).

Directional Cues Given in Virtual Collision

There are a total of two directional cues given to the user in the process of moving a hand for a virtual collision. The first is the *unit normal vector* ($\vec{u}_n$) of the palm surface. The second is the *unit velocity vector* ($\vec{u}_v$) of the hand. The user will be able to estimate the collision direction separately from each of these cues (see Fig. 2). The friction between the ball and the palm also affects the direction of the impact force and the user can estimate the friction from the visual perception of the material texture of the ball. However, such visual estimation is assumed to be unreliable compared to the two kinesthetic cues described earlier and is therefore not considered in the implementation of the technique.

If the functions representing the user's estimation process from each cue vector are $f_n()$ and $f_v()$, the unit collision direction vectors estimated by the user from each cue ($\hat{\vec{u}}_n$ and $\hat{\vec{u}}_v$) can be expressed as follows:

$$\hat{\vec{u}}_n = f_n(\vec{u}_n) \quad \text{and} \quad \hat{\vec{u}}_v = f_v(\vec{u}_v) \tag{1}$$

Note that the hat notation (ˆ) in the vectors means that the value is estimated by the user through the perception process, so they are latent variables that cannot be measured directly.

Mean and Variance of Estimates

The user's estimates of collision direction ($\hat{\vec{u}}_n$ and $\hat{\vec{u}}_v$) are probabilistic variables. This section models the mean and variance of each estimate.

Mean of Estimates

In psychophysics, Stevens' law [67] predicts that there is a power function relationship between the magnitude of the actual physical quantity and the perceived physical quantity. However, it is known that the exponent of the power function is close to 1.0 for proprioception [39, 59, 60] or vision [40, 13, 64] on distance and speed perception. Therefore, linearity can be assumed in the user's perception of distance and speed.

The estimation of hand orientation and velocity requires hand distance and speed estimation for each axis in three-dimensional space. From the linearity of distance and speed perception, we can assume that unit vectors representing the collision direction estimated by the user are unbiased:

$$\hat{\vec{\mu}}_n = \mathrm{E}[\hat{\vec{u}}_n] = \mathrm{E}[\vec{u}_n] \quad \text{and} \quad \hat{\vec{\mu}}_v = \mathrm{E}[\hat{\vec{u}}_v] = \mathrm{E}[\vec{u}_v] \tag{2}$$

This means that the mean of the collision direction estimated by the user ($\hat{\vec{u}}_n$ or $\hat{\vec{u}}_v$) is equal to the mean direction of the actual physical cue vectors ($\vec{u}_n$ or $\vec{u}_v$).

Variance of Estimates

The higher the variance of the estimate, the less reliable it is. To model the variance of the user's estimated vectors ($\hat{\vec{u}}_n$ and $\hat{\vec{u}}_v$), we make one heuristic assumption: if the hand is moving fast, the user's estimation of the velocity vector of the hand ($\hat{\vec{u}}_v$) will be more reliable than the estimation of the palm's normal vector ($\hat{\vec{u}}_n$). We assume this because if the hand speed is very fast, the palm's normal vector will change very quickly over time compared to the change in the hand's velocity vector. This does not give the user enough time to estimate the normal vector of the palm. According to the drift-diffusion model of the reaction time [57, 56], the reliability of the user's perception decreases when the user is not given enough time to perceive the state of the surroundings [34].

This assumption can be expressed as a function of the movement speed of the hand (s) as follows:

$$\hat{\sigma}_n = a_n \cdot \exp(b_n \cdot s) \;\text{ and }\; \hat{\sigma}_v = a_v + 1/(\exp(b_v \cdot s) - 1) \tag{3}$$

Equations above tell us that when the hand speed is very fast ($s \to \infty$), the variance of the user's estimation of $\vec{u}_n$ (i.e., $\hat{\sigma}_n^2$) is greatly increased. In contrast, when the hand speed is very slow ($s \to 0$), the user's estimation of the velocity vector $\vec{u}_v$ has a relatively higher variance (i.e., $\hat{\sigma}_v^2$) than the perception of the normal vector. The a and b parameters are positive free parameters to be determined for each individual user or each user population through a calibration experiment. The calibration process is covered in detail in Study 1.

Integration of Perceived Directions

The cue integration theory [16] provides a statistical model of human perception. In the model, it is assumed that humans try to estimate a specific physical quantity S from the environment. For example, in the case of a virtual collision, the physical quantity can be the direction vector of the collision response. The cues that allow estimation of the physical quantity can then be given from several sensory channels simultaneously. Let the physical quantity estimated from i-th sensory channel be $\hat{S}_i$. This can be expressed from the perceptual estimation function f_i of the i-th sensory channel as follows:

$$\hat{S}_i = f_i(S) \tag{4}$$

Each perception is a probabilistic value and therefore has a specific mean ($\hat{\mu}_i = \mathrm{E}[\hat{S}_i]$) and variance ($\hat{\sigma}_i^2 = \mathrm{Var}[\hat{S}_i]$). At this time, the cue integration theory assumes that the process of human integrating different estimates into a final one ($\hat{S}$ is statistically optimal [16]. Without prior probability of each estimate, this integration process can be expressed as the maximum likelihood estimation (MLE) as follows:

$$\hat{\mu} = \mathrm{E}[\hat{S}] = \sum_i w_i \hat{\mu}_i \;\text{ with }\; w_i = \frac{1/\hat{\sigma}_i^2}{\sum_j 1/\hat{\sigma}_j^2} \tag{5}$$

$\hat{\mu}$ is the mean of the integrated estimate and w_i is the weight for the i-th estimate during the integration process. As shown in the equation, we can see that humans give smaller weights to the less reliable (or higher variance) cues in the integration process. As a result, the variance of the integrated estimate ($\hat{\sigma}^2$) is lower than the variance of the individual estimates. The following shows the variance ($\hat{\sigma}^2$) of the integrated estimate when two different estimates exist for a physical quantity:

$$\hat{\sigma}^2 = \hat{\sigma}_1^2 \hat{\sigma}_2^2 / (\hat{\sigma}_1^2 + \hat{\sigma}_2^2) \tag{6}$$

Integrated Perception of Collision Direction

Equations 2 and 3 mathematically represent the mean ($\hat{\vec{\mu}}_n$ and $\hat{\vec{\mu}}_v$) and variance ($\hat{\sigma}_n^2$ and $\hat{\sigma}_v^2$) of two different estimates of the user about the direction of the collision. Based on the cue integration theory, we can explain the process by which the user integrates two different estimates into one final estimate of the collision direction. The user is considered to be a statistically optimal encoder based on MLE:

$$\begin{gathered}\vec{\mu} = \mathrm{E}[\hat{\vec{u}}] = w_n \vec{\mu}_n + w_v \vec{\mu}_v \\ \text{where } w_n = \frac{1/\sigma_n^2}{1/\sigma_n^2 + 1/\sigma_v^2} \;\text{ and }\; w_v = \frac{1/\sigma_v^2}{1/\sigma_n^2 + 1/\sigma_v^2}\end{gathered} \tag{7}$$

This predicts that as the hand speeds up, users will estimate the collision direction with a higher weight on the hand velocity vector ($\vec{u}_v$) than the normal vector of the palm ($\vec{u}_n$).

CUE INTEGRATION COLLISION TECHNIQUE

This study proposes a novel collision response technique based on the user perception model derived from the previous sections. The technique generates a response of the collision in the same direction as the one estimated by the user. If the direction of the collision response generated from the system is $\vec{u}_r$ (see Fig. 2), it can be determined as follows:

$$\vec{u}_r = w_n \vec{u}_n + w_v \vec{u}_v \tag{8}$$

$\vec{u}_v$ and $\vec{u}_n$ can be tracked using motion capture or other hand tracking techniques (see Fig. 1). w_n and w_v are weights defined in Equation 7, which depend on the speed of the hand and can be calculated for each frame when the a_n,a_v, b_n, b_v values of Equation 3 are known in advance through a calibration study.

From the assumption of unbiased collision direction estimation (see Equation 2), this technique makes the average direction of the collision response equal to the average of the directions the user estimates.

$$\mathrm{E}[\vec{u}_r] = w_n \mathrm{E}[\vec{u}_n] + w_v \mathrm{E}[\vec{u}_v] = w_n \hat{\vec{\mu}}_n + w_v \hat{\vec{\mu}}_v = \hat{\vec{\mu}} \qquad (9)$$

Calibration of the Model Parameters

Basically, in order to determine the free parameters of the model (a_n,a_v, b_n, b_v), we have to experimentally measure the variance of the user's estimates ($\hat{\sigma}_n^2$ and $\hat{\sigma}_v^2$) and then fit it into Equation 3. However, perceptual variances are latent variables that are difficult to measure directly through experimentation. Instead, we can design an experiment to get the parameters indirectly. The experiment assumes that if the collision response is performed in the same direction as the user estimated (as in Equation 9), the variance of the collision response vector itself ($\mathrm{Var}[\vec{u}_r]$) will also be minimized. Based on this assumption, we record the collisions between the user and the virtual object in various directions and sizes, and then optimize the a_n, a_v, b_n, b_v values to minimize $\mathrm{Var}[\vec{u}_r]$ for all of the trials. The design and implementation of the actual calibration experiment are described in Study 1.

Implementation in Commercial Physics Engines

The proposed collision technique can be easily put to practical use by modifying the *collision response pipeline* of existing physical engines. Existing physics engines return the direction and magnitude of impact force as a collision response. Our collision technique maintains the magnitude of the impact force returned by the physics engine, but changes its direction as calculated in Equation 8. Algorithm 1 shows the pseudocode of our collision technique. In the pseudocode, the magnitude of the impact force ($\vec{I}$) returned by the collision detection pipeline (collisionDetected) of the physics engine remains the same, but the direction changes to the direction estimated by the user.

Most physics engines provide collision response callbacks in a similar manner. For example, `Unity`, the most popular game development platform [43], returns an impact force vector from the `onCollsionEnter` callback function the first time a collision is detected and from the `onCollisionStay` function while the collision continues.

As shown in the last line of the pseudocode, other factors affecting the direction of the collision, such as friction, should not be reflected in the impact force vector before our technique is applied. If so, the effect is overwritten by our technique.

Algorithm 1 Cue Integration Collision Technique

$\vec{I}, \vec{u}_n, \vec{u}_v, s, \sigma_n, \sigma_v, w_n, w_v, \vec{u}_r \leftarrow 0$
while $\mathit{CollisionDetected} = \mathit{true}$ **do**
  $\vec{I} \leftarrow$ Impulse vector returned by physics engine
  $\vec{u}_n, \vec{u}_v \leftarrow$ Unit normal and velocity vector of user's palm
  $s \leftarrow$ Speed of user's hand
  $\sigma_n^2 \leftarrow \mathit{getPerceptionNormalVariance}(s)$
  $\sigma_v^2 \leftarrow \mathit{getPerceptionVelocityVariance}(s)$
  $w_n \leftarrow \mathit{getNormalVectorWeight}(\sigma_n^2, \sigma_v^2)$
  $w_v \leftarrow \mathit{getVelocityVectorWeight}(\sigma_n^2, \sigma_v^2)$
  $\vec{u}_r \leftarrow w_n \cdot \vec{u}_n + w_v \cdot \vec{u}_v$
  $\vec{I} \leftarrow |\vec{I}| \cdot \vec{u}_r$
  $\vec{I} \leftarrow \mathit{applyFrictionEffect}(\vec{I})$

STUDY 1: CALIBRATION OF THE MODEL PARAMETERS

The Study 1 aims to determine the values of the free parameters in our proposed collision perception model (see a_n, a_v,b_n, and b_v in Equation 3). Participants were required to collide their main hand in various directions and speeds towards a virtual ball (see Fig. 3). The analysis then finds those parameter values that minimize the variance of the $\vec{u}_r$ vector defined in Equation 8. This calibration procedure is described in the Calibration of the Model Parameters section.

Method

Participants and Design

We recruited 16 participants (9 females, 7 males). Their average age was 23.5 (SD=2.89), all without glasses and all right-handed. Their reported familiarity with the head-mounted display (HMD) they reported was 2.81 (SD=2.10) on a 7 point scale, and their familiarity with the interaction with virtual objects was 2.81 (SD=1.84). They received a gift card of 10 $ as a reward for their participation.

The experiment followed a $6 \times 3 \times 2$ within-subject design with three independent variables:

- *Ball Direction*: Forward, Backward, Right, Left, Up, and Down
- *Ball Speed*: 1, 2.5, and 6 m/s
- *Ball Diameter*: 0.2 and 0.07 m

Each *Ball Direction* condition was set based on the orientation of the HMD worn by the user (see Fig. 3).

Task

The participant wore an HMD on the head and a rectangular wooden board on their primary hand. Each finger of the participant was fastened to the wooden board with Velcro (see Fig. 4). Then a virtual ball was shown in front of the participant's head. One second after it was created, the ball started to move in the given *Ball Speed* condition and *Ball Direction* condition. After traveling two meters, the ball returned to its original position and stopped. The participant who observed the movement hit the ball by hand when the ball stopped. The participant's main hand was represented in virtual space as a board with the same size and thickness as the wooden board (see Fig. 3). At this point, the participant was required to strike the ball in such a way that the motion of the ball just observed could be reproduced by the collision. Because the ball did not actually move from the collision, the participant had to imagine the consequences of the collision. Participants were not given any other instructions on how to strike the ball. One second after the collision was complete and the hand was

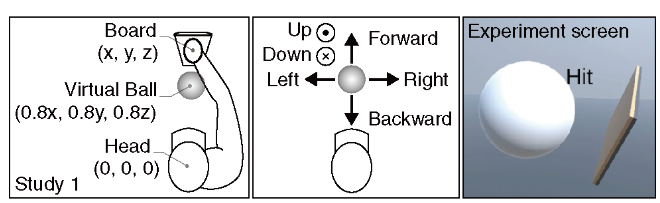

Figure 3. Experimental settings specific to Study 1

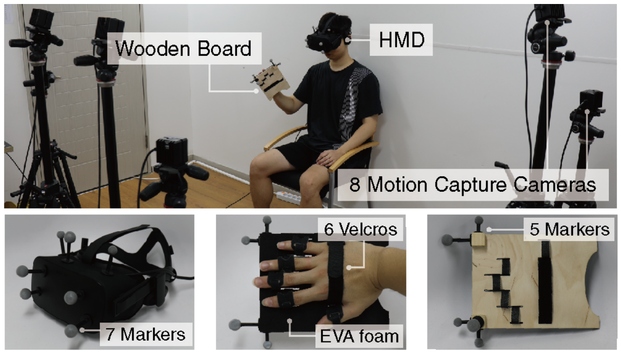

Figure 4. Common experimental settings used for Study 1 and Study 2

removed from the ball, the trial ended and a new trial began with new speed and direction conditions.

Apparatus

The experimental application was run on a Windows desktop PC (64-bit Windows10, Intel Core i7-9700K CPU @ 3.60 GHz, 32 GB RAM, and NVIDIA GeForce GTX 1070 Ti). The position and orientation of the HMD (`Oculus Rift`, 1.38) and the wooden board (20 cm×15cm×1cm, 117 grams) worn on the hand were tracked from a separate motion capture system (`Optitrack Prime 17W`, eight cameras). On the HMD and the board, seven and five markers were attached, respectively, as shown in Fig. 4. The group of markers attached to each object was set up as a single rigid body in motion capture software. In order to verify that the established rigid bodies successfully represent the positions of the HMD and the wooden board, the actual positions of the HMD and the wooden board are compared with the positions of the rigid bodies in the `Unity` scene. As a result, the position error of the rigid body of the HMD and the wood board was 1.5 mm (SD= 0.25) and 0.9 mm (SD=0.13), respectively. For making free movement of the wrist, the underside of the wooden board was cut as an ellipse shape. The frame rate of the HMD was maintained at 90Hz, and the sampling rate of motion capture was maintained at 240Hz. The final experimental application was implemented through `Unity` (2019.1.0f2).

Setup and Procedure

The motion capture system was calibrated each time just before the experiment. Participants sat in a comfortable chair and filled out a consent form and pre-questionnaire. Then, after wearing the HMD and the wooden board, the participants were instructed to stretch their arms out in front of the head for 10 seconds. At this time, the original position of the ball was determined at the ratio of 8 to 2 between the mean position of the HMD and board (see Fig. 3). After a practice trial was performed once for each unique condition (36 times in total), they performed the main experiment. The experiment gave the *Ball Speed*, *Ball Direction*, and *Ball Diameter* conditions in random order, and one unique condition consisted of 20 repeated trials (a total of 720 trials per participant). The experiment took about one hour per participant. We logged the position, velocity, and orientation of the HMD, wooden board, and virtual ball along with time stamps during the experiment.

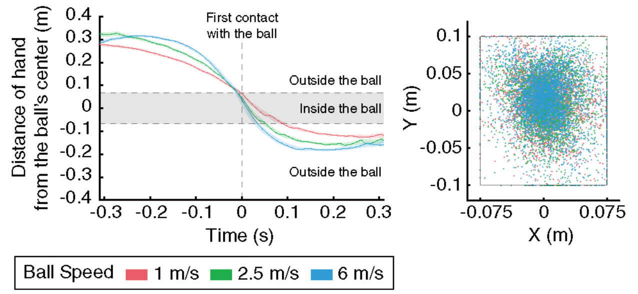

Figure 5. The change in distance between the center point of the participant's hand and the center of the virtual ball when the contact start timing is t = 0 (left), the hit points of the ball on the wooden board (right)

Results

In total, 11,520 virtual collisions were logged from 16 participants. For more accurate calibration, collision detection was performed directly by analyzing the motion capture data regardless of the collision detection events from the physics engine of `Unity`. The participant's hand and ball were considered to collide while the distance between the rigid body of the wooden board and the center of the ball was less than the radius of the ball.

During the collision, we calculated the unit normal vector of the participant's palm ($\vec{u}_n$) from the orientation of the wooden board rigid body. In addition, the unit velocity vector of the participant's hand ($\vec{u}_v$) and the moving speed of the user's hand (s) can be calculated from the velocity vector of the wooden board. The velocity vector is obtained by numerically differentiating the trajectory of the center point of the plate.

Outlier Removal

Based on the $\vec{u}_n$ and $\vec{u}_v$ measured at the moment the hand and ball first collided, and the angular error between the vectors and the given *Ball Direction* vector, we defined and removed the outliers. We considered trials where the angle error exceeded the three median absolute deviations (3MAD) [36], either $\vec{u}_n$ or $\vec{u}_v$, as outliers. As a result, 628 trials, 5.45% of the total, were removed and the subsequent analysis was performed on the remaining 10,892 trials.

Descriptive Statistics

There was no specific instruction on how to strike the ball, but all the participants struck the ball in a similar manner. They mainly touched the ball near the point of their finger on the wooden board (see Fig. 5, right). They mainly moved their hands to the point beyond the center of the ball (see Fig. 5 left). The speed of the center point of the wooden board measured at the first contact of the ball and hand was 1.31 (SD=0.62), 1.95 (SD=0.83), and 2.55 (SD=1.10) m/s when the target *Ball Speed* was 1, 2.5, and 6 m/s, respectively. The $\vec{u}_n$ and $\vec{u}_v$ measured at the first contact of the ball and hand showed an angular error of 22.38° (SD=15.48) degrees and 20.66° (SD=13.52) degrees, respectively, on the basis of the given *Ball Direction* vector.

Model Fitting

As described earlier in the Calibration of the Model Parameters section, we looked for model parameters that minimize the variance of the integrated collision response vector $\vec{u}_r$. More specifically, this process can be formulated as the following

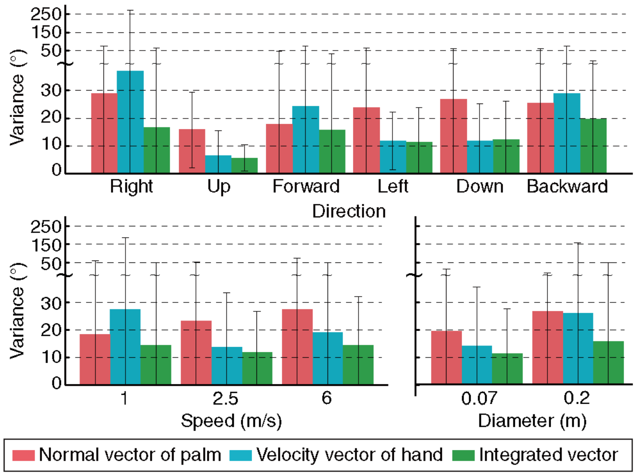

Figure 6. The angular variances of the $\vec{u}_r$ vector (green) obtained by integrating $\vec{u}_n$ (red) and $\vec{u}_v$ (blue) are lower than the respective angular variances that are not integrated.

optimization problem:

$$\underset{a_n,a_v,b_n,b_v}{\text{minimize}}\ \mathrm{Var}[\vec{u}_r] = \mathrm{Var}[w_n\vec{u}_n + w_v\vec{u}_v]$$
$$\text{subject to } w_n = \frac{1/\sigma_n^2}{1/\sigma_n^2 + 1/\sigma_v^2}\,,\ w_v = \frac{1/\sigma_v^2}{1/\sigma_n^2 + 1/\sigma_v^2} \tag{10}$$
$$\sigma_n = a_n \exp(b_n s)\,,\ \sigma_v = a_v + 1/(\exp(b_v s) - 1)$$

$\vec{u}_n$, $\vec{u}_v$, and s are the unit normal vector of the palm, the unit velocity vector of the palm, and the hand speed, which are obtained at the first contact of the ball with the wooden board, and can be obtained for each collision trial. The variance was calculated as the variance of the *angular error* with respect to the mean vector. For all 10,892 trials, optimization was performed using the `patternsearch` algorithm provided by `Matlab`'s global optimization toolbox. The resulting parameters are as follows: a_n=1.02, a_v=2.59, b_n=0.75, and b_v=173.32.

Discussion

By substituting the four parameters we just obtained into our model (Equations 3 and 7), we can calculate how much variance the individual estimates have when the participants estimate the normal vector of the palm ($\vec{u}_n$) and the velocity vector of the hand ($\vec{u}_v$). Figure 7 shows the result. In the figure, our model explains that as the hand speed increases, the variance of the estimate of the normal vector increases, but the variance of the estimate of the velocity vector remains relatively constant. According to the cue integration theory, participants assign higher integration weights to estimates with lower variance. As a result, the right side of Fig. 7 shows the integration weights (w_n and w_v) calculated from the MLE. As the figure shows, participants will estimate the collision direction almost entirely on the hand's velocity vector when the hand speed begins to exceed 3 m/s.

After optimization, the variance of the integrated vector (M=13.5, SD=24.4) was lower than that of the individual vectors $\vec{u}_n$ (M=23.1, SD=32.2) or $\vec{u}_v$ (M=20.1, SD=92.8). This effect was similarly observed for different ball diameters, speeds, and directions (see Fig. 6).

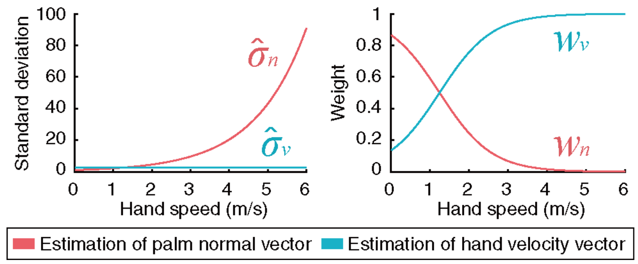

Figure 7. When a user estimates palm normal and velocity vectors, each variance changes as a function of hand speed (left), Weights between normal and velocity vectors varying as a function of hand speed (right)

STUDY 2: EVALUATION OF COLLISION TECHNIQUE

We implemented the collision technique we proposed in Algorithm 1 using the parameters obtained from Study 1. Study 2 compares the accuracy and precision of the collision response produced by our collision technique with a baseline. In the experiment, participants strike a virtual ball by hand. During the strike, the virtual ball was forced and moved. Participants were asked to strike the ball so that it hits as close as possible to the distant target point (see Fig. 8).

Method

Participants and Design

We recruited 18 participants (13 males, 5 females). Their average age was 23.7 yrs (SD=3.11), all without glasses and all right-handed. They reported their familiarity with the HMD as 2.11 (SD=1.24) on a 7 point scale, and their familiarity with the interaction with virtual objects was 2.11 (SD=1.05). They received a gift card of 10 $ as a reward for their participation. The experiment followed a $2\times2\times2\times2$ within-subject design with four independent variables:

- *Collision Technique*: Conventional Response or Integrated Response
- *Target Speed*: Slow (collision speed lower than 4 m/s) or Fast (collision speed higher than 4 m/s)
- *Friction*: 0.0 or 0.5
- *Bounciness*: 0.0 or 0.5

Collision Technique: This represents two different techniques for generating a collision response. The Conventional Response condition represents the collision response technique provided by `Unity`, the most widely used commercial 3D development platform. `Unity`'s physics engine is based on `NVIDIA`'s `PhysX` engine, which is also the most widely used gaming physics engine [45] and the highest rated physics engine overall [61, 22, 27, 6, 46]. We set the collision detection mode in `Unity` to *continuous speculative* to maximize its performance. Otherwise, we used the default physics setting (2 bounce threshold, 0.005 sleep threshold, 0.01 contact offset, 6 solver iteration, 0.012 fixed timestep, and 1 default solver velocity iterations). The collision response was then automatically generated from the system.

The Integration Response condition represents our proposed collision response technique and is implemented using the parameters obtained from Study 1 as described in Algorithm 1. The technique was implemented using `Unity`'s physics engine pipelines. If a collision occurs from the physics engine, the impact force is returned via `onCollisionEnter` or `onCollisionStay` callback function in `Unity`. Within

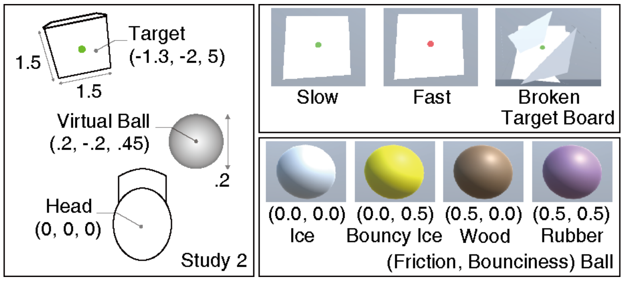

Figure 8. Experimental settings specific to Study 2

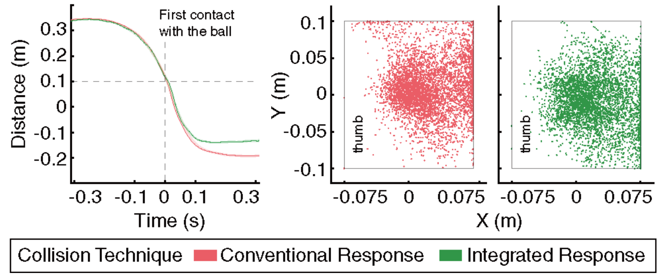

Figure 9. The change in distance between the center point of the participant's hand and the center of the virtual ball when the contact started timing is t = 0 (left), the hit points of the ball on the wooden board (right)

the callback function, the Integration technique maintains the magnitude of the impact force but changes its direction to the cue integrated vector ($\vec{u}_r$) as in Equation 8. After that, the movement of the object is processed by the physics engine based on the updated impact force.

Note) The effect of friction in the `Unity` physics engine is added after all processes in the collision callback function have finished. The bounciness only affects the magnitude of the impact vector. Thus, friction and bounciness effects are equally applied regardless of the type of *Collision Technique*.

Target Speed: In order to compare the performance of the two *Collision Techniques* over a wide range of speeds, participants were required to control the speed of the ball hitting the target at two levels (fast or slow). The criterion of 4 m/s was determined by a pilot experiment.

Friction and *Bounciness*: `Unity`'s physics engine can handle friction and elastic effects between two colliding objects. In the material settings panel, users can determine the degree of friction and bounciness by adjusting parameter values between 0 (no effect) and 1 (maximum effect). To compare the two techniques in a wide range of situations, friction and bounciness were controlled in two steps: 0.0 and 0.5. Intuitively, if the friction parameter is 0.0, the frictional effect is similar to that of an ice surface and 0.5 is similar to that of a wooden block. If the bounciness parameter is 0.0, it can be considered as a very hard ball, and if it is 0.5, it is a rubber ball. For each condition, the material properties of the ball and the user's avatar were set identically. Participants were given information about the material of each ball, and each ball was created in a different color (see Fig. 8).

The dependent variables are the accuracy and precision of the participant's target hit. For statistical testing, we used the repeated-measure ANOVA with an α-level of 0.05.

Task and Apparatus

The experimental environment using HMD and wooden board in Study 2 was set up the same as in Study 1, except the firmware of the HMD was updated (1.40.1). When the experiment begins, a virtual ball is created which is stopped in front of the participant. Also farther away, a target plate of a certain size (1.5m × 1.5m) is placed facing the participant (see Fig. 8). At the center of the target plate is a fixed target point that the player must hit with the ball. The color of the target point was green in the Slow condition and red in the Fast condition. When the participant strikes the ball, the impact force is calculated using the given *Collision Technique* and the ball moves according to the force. If the speed at which the ball hits the target plate matches the given *Target Speed* condition, the target plate is broken and a new ball is created for the next trial. All trials that did not meet the ball's target speed, whether they hit or missed the target plate, were discarded. If the ball went 7 meters away from the origin, or moved more than 5 seconds after being hit by hand, the trial was considered a failure and the ball returned to its original position and restarted. Participants were asked to aim as accurately as possible.

Setup and Procedure

The motion capture system was calibrated each time just before the experiment. Participants sat in a comfortable chair and filled out a consent form and pre-questionnaire. Then after wearing the HMD and the wooden board, the participants waited for 10 seconds to determine the baseline HMD position. Participants first experimented with one of the *Collision Technique* conditions. At the end of the first *Collision Technique*, we experimented with another one, which was counter balanced for each participant. Within a *Collision Technique* condition, participants shot the ball 30 times for each *Target Speed*, *Friction*, and *Bounciness* condition (240 time in one Collision Technique). Before that, a total of 40 practice shots were given, five times for each condition. The order of each condition was randomized. After experimenting with a *Collision Technique*, the participants wrote a NASA-TLX [23] report and subjective ratings on the technique. We logged data related to collisions and rigid bodies.

Results

A total of 9,860 collisions were logged, including 1,220 failure trials. Participants hit the ball with movements similar to Study 1 (see Fig. 10). Participants' average hand speed during the collision was 2.149 m/s (SD=1.684) in all trials: 1.143 m/s (SD=0.658) in Slow condition and 3.499 m/s (SD=1.696) in Fast condition. In addition, the average speed of the hand during the collision was 2.154 m/s (SD=1.715) in Conventional Response condition and 2.145 m/s (SD=1.652) in Integrated Response. The average time the participants' hands were in contact with the ball was 0.108 ms (SD=0.051) in all trials: 0.104 s (SD=0.051) in Conventional Response condition and 0.112 s (SD=0.051) in Integrated Response condition. After the collision with the hand, the average speed of the ball was 4.739 m/s (SD=3.134) in all trials: 2.245 m/s (SD=0.749) in Slow condition and 7.036 m/s (SD=2.711) in Fast condition.

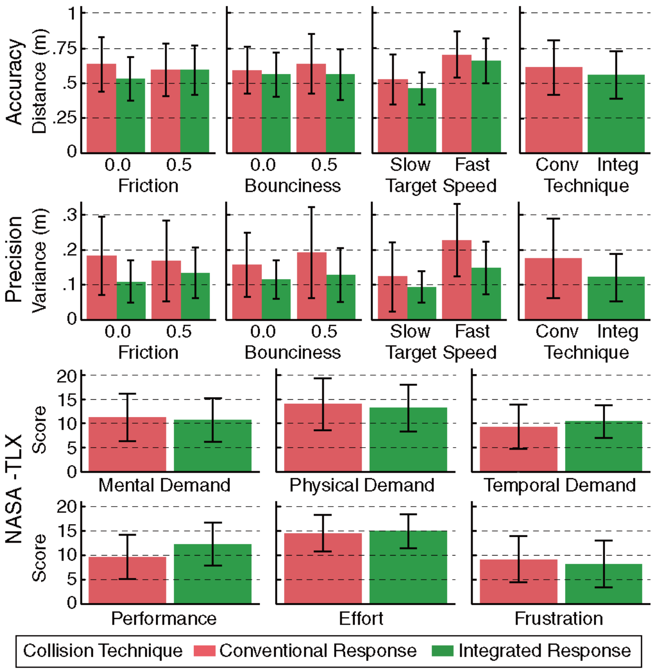

Figure 10. Results of Study 2: The Integrated Response technique improves the accuracy of virtual collisions by 8.77%, precision by 30.26% (inclusive of all trials), and subjective performance by 26.8%.

The average magnitude of the impact on the ball from the collision was 1.561 N (SD=1.956) in all trials: 1.677 N (SD=2.181) in Conventional Response condition and 1.448 N (SD=1.767) in Integrated Response condition.

Accuracy and Precision

The position in three-dimensional space where the ball hit the target plate was converted to a two-dimensional plane coordinate system of the target plate. The position of the target point in this coordinate system is (0,0) in meters (see Fig. 11). The horizontal coordinates on the plate are represented by the x-axis, and the vertical coordinates on the plate are represented by the y-axis. Accuracy is defined as the root-mean-square deviation (RMSD) of the distance from (0,0) to the point where the ball hit, and precision is defined as the standard deviation of the hit point distribution in the x and y directions. To simplify the analysis, precision was calculated as the average of the x and y variances. If the coordinates of the point where the ball hit in the i-th trial are (x_i, y_i), the accuracy and precision are expressed as follows:

$$\text{Accuracy} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i^2 + y_i^2)} \quad \text{[all unit: meters]}$$

$$\text{Precision} = \frac{1}{2}\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x}_i)^2} + \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}\right)$$

N is the total number of trials, and $\bar{x}_i$ and $\bar{y}_i$ are the average of the x and y coordinates of the hit points, respectively.

Effect of Collision Technique on Accuracy: The effect of the *Collision Technique* on the shooting accuracy was not statistically significant ($F(1,17)$=2.248, p=0.152, η_p^2=0.117). The accuracy was 0.491 (SD=0.092) in Conventional Response conditions and 0.477 (SD=0.102) in Integrated Response conditions. This indicates that the Integrated Response has improved collision accuracy by 2.85 %. The interaction effect between *Collision Technique* and *Target Speed* on shooting accuracy was not significant ($F(1,17)$=2.573, p=0.127, η_p^2=0.131).

There was a significant interaction effect between *Collision Technique* and *Friction* on shooting accuracy ($F(1,17)$=6.638, p=0.02, η_p^2=0.281). From the pairwise comparison, a significant difference between Conventional Response (M=0.498, SD=0.089) and Integrated Response (M=0.465, SD=0.104) was observed when the friction was 0.0 (p=0.01). This means that without friction, the Integrated Response has improved shooting accuracy by 6.49 %. If the friction was 0.5, the difference between the two techniques was not significant (p=0.731). At that time, the mean accuracy of Conventional Response and Integrated Response was 0.483 m (SD=0.094) and 0.488 m (SD=0.099), respectively. The interaction effect between *Collision Technique* and *Bounciness* on shooting accuracy was not significant ($F(1,17)$=1.425, p=0.249, η_p^2=0.077).

Effect of Collision Technique on Precision: The effect of the *Collision Technique* on the shooting precision was statistically significant ($F(1,17)$=23.788, $p<0.001$, η_p^2=0.583). The precision was 0.103 (SD=0.038) in Conventional Response conditions and 0.082 (SD=0.027) in Integrated Response conditions. This indicates that the Integrated Response has improved collision precision by 20.39 %.

The interaction effect between the *Collision Technique* and *Target Speed* on shooting precision was also significant ($F(1,17)$=30.12, $p<0.001$, η_p^2=0.639). From the pairwise comparison, a significant difference between Conventional Response (M=0.126, SD=0.030) and Integrated Response (M=0.089, SD=0.027) was observed when the target speed was Fast ($p<0.001$). In this case, the Integrated Response has improved shooting precision by 28.81 %. If the target speed was Slow, the difference between the two techniques was not significant (p=0.142). In this case, the mean precision of Conventional Response and Integrated Response was 0.081 m (SD=0.032) and 0.076 m (SD=0.024), respectively.

There was a significant interaction effect between *Collision Technique* and *Friction* on shooting precision ($F(1,17)$=9.204, p=0.007, η_p^2=0.351). From the pairwise comparison, a significant difference between Conventional Response and Integrated Response was observed when the friction was 0.0 ($p<0.001$) or 0.5 (p=0.011). When friction was 0.0, the mean precision of the Conventional Response and Integrated Response were 0.106 m (SD=0.038) and 0.079 m (SD=0.028) (25.71 % improvement), respectively, and 0.100 m (SD=0.038) and 0.086 m (SD=0.025) (21.36 % improvement) respectively when the friction was 0.5. The interaction effect between *Collision Technique* and *Bounciness* on shooting precision was not significant ($F(1,17)$=0.078, p=0.783, η_p^2=0.005).

Including Failed Trials: The effect of the *Collision Technique* on the number of failed trials was statistically significant ($F(1,17)$=6.503, p=0.021, η_p^2=0.277). The average number of failed trials in Conventional Response was 4.99 (SD=5.43)

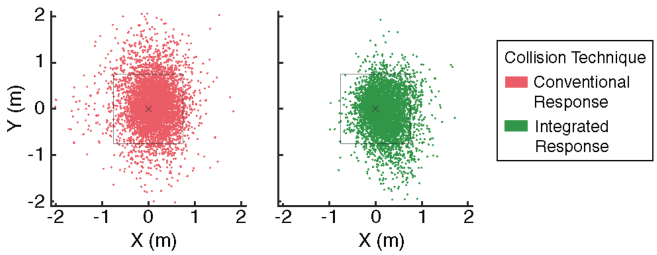

Figure 11. The ball hit points on the target plate, target point: (0,0)

and the average number of failed trials in the Integrated Response was 3.48 (SD=3.43). This means that in the Integrated Response condition, participants failed about 30.26 % less than the Conventional Response condition.

We further test the effect of the *Collision Technique* on accuracy and precision, including failed trials. We assumed that the target plate was wider (10 m × 10 m) than it actually was and found the contact points that would be hit if the ball continued to move in failed trials. Trials that did not collide with the widened plate were still removed. The effect of the *Collision Technique* on shooting accuracy ($F(1,17)$=10.383, p=0.005, η_p^2=0.379) and shooting precision ($F(1,17)$=21.332, $p<0.001$, η_p^2=0.557) was statistically significant, including the expected hit points of failed trials. The accuracy (M=0.562, SD=0.169) and precision (M=0.122, SD=0.068) of the Integrated Response is improved by 8.77% and 30.29%, respectively, compared to the accuracy (M=0.616, SD=0.192) and precision (M=0.175, SD=0.114) of the Conventional Response.

Performance in Physically Realistic Conditions: Situations with zero friction or bounciness are difficult to encounter in reality. Comparing the performances of the techniques only when the friction and bounciness were both 0.5, the accuracy was 0.635 (SD=0.222) and 0.567 (SD=0.192) in the Conventional and Integrated conditions (10.7% improvement), and the precision was 0.201 (SD=0.147) and 0.131 (SD=0.085), respectively (34.8% improvement).

Workload Index

The scores for the Conventional Response condition are as follows: mental demand (M=11.1,SD=4.7), physical demand (M=13.8, SD=5.1), temporal demand (M=9.2, SD=4.4), performance (M=9.7, SD=4.4), effort (M=14.5, SD=3.6), frustration (M= 9.2,SD=4.6). For the Integrated Response condition: mental demand (M=10.5,SD=4.3), physical demand (M=13,SD=4.6), temporal demand (M=10.2, SD=3.2), performance (M=12.3, SD=4.3), effort (14.9, SD=3.4), frustration (M=8.2, SD=4.7). From the paired sample t-test, the performance score of the Integrated Response was significantly higher ($t(17)$=2.31, p=0.034, d=0.54) than the Conventional Response (26.8 % improvement). The difference between the two techniques in other scores was not significant ($p>0.26$).

Subjective Rating

Eight participants commented that the Integrated Response was more manageable, comfortable, easy, and less hard than the Conventional Response. Three participants commented the Conventional Response was easier, however, the accuracy and precision of the Integrated Response of two of those three have improved over the Conventional Response. The remaining of the three participants who gave negative feedback had higher precision and slightly lower accuracy (Conventional: 0.637 m, Integrated: 0.647 m) in the Integrated Response condition. Some notable comments were, "The ball hitting (in Integrated Response condition) is more realistic (than the Conventional Response)" and "If there is gravity, this model (Integrated Response) can be used for a baseball game".

Discussion

Our collision method has been shown to have better accuracy, precision and subjective performance score than the conventional baseline. In particular, the improvement in precision (30.26 %) was greater than the improvement in accuracy (8.77 %), because our model focused on the process by which the variance of the direction perception itself is reduced by the user's MLE, not an error problem between a particular ground truth direction vector and the user's perceived direction vector.

LIMITATIONS AND CONCLUSION

This study proposed a novel collision technique to improve the directional accuracy and precision of virtual collision responses in VR. It identified the cues that allow the user to estimate the direction of a virtual collision. After constructing mathematical models for the variances of the direction estimates from each cue, this study inferred the user's final estimate of the collision direction based on cue integration theory. In a commercial physics engine, by setting the collision response direction equal to the inferred collision direction, the accuracy and precision of the virtual collision could be improved by 8.77 % and 30.26 %, respectively.

The proposed collision technique has the advantage that it does not require any additional hardware, is simple to calculate, and can be directly applied to existing systems, but it also has some limitations. First, this study was performed assuming a virtual avatar of a user with a low degree of freedom, such as a plate attached to a hand. This means that it is not known whether the cue-integration collision technique can be applied to the high degree of freedom of body skeletons measured from today's advanced computer vision technologies. We envision that for such a high degree of freedom user avatars, the number of free parameters in the perceptual model will increase exponentially, and therefore data-driven machine learning will need to be applied, rather than a simple calibration as in Study 1.

Second, this study does not guarantee whether the collision technique proposed in this study will still work if simple haptic methods, such as vibrotactile actuators [10] or electric muscle activation [37, 38], are also applied to the user. Above all, we do not know how the additional sensory signals from such devices affect the user's perception of virtual collisions. From simple experiments, the effects of such devices on the user's collision accuracy, precision, and even sense of agency [31], should be verified. In the process, we expect that the model presented by this study could provide useful insights.

ACKNOWLEDGEMENTS

This research was funded by National Research Foundation of Korea (2020R1A2C4002146) and Korea Creative Content Agency (R2019020010).

REFERENCES

[1] Ferran Argelaguet and Carlos Andujar. 2008. Improving 3D selection in VEs through expanding targets and forced disocclusion. In *International Symposium on Smart Graphics*. Springer, 45–57.

[2] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136.

[3] Jeremy N Bailenson and Nick Yee. 2006. A longitudinal study of task performance, head movements, subjective report, simulator sickness, and transformed social interaction in collaborative virtual environments. *Presence: Teleoperators and Virtual Environments* 15, 6 (2006), 699–716.

[4] Patrick Baudisch, Douglas DeCarlo, Andrew T Duchowski, Wilson S Geisler, and others. 2003. Focusing on the essential: considering attention in display design. *Commun. ACM* 46, 3 (2003), 60–66.

[5] Hrvoje Benko and Steven Feiner. 2007. Balloon selection: A multi-finger technique for accurate low-fatigue 3d selection. In *2007 IEEE Symposium on 3D User Interfaces*. IEEE.

[6] Adrian Boeing and Thomas Bräunl. 2007. Evaluation of real-time physics simulation systems.. In *Graphite*, Vol. 7. 281–288.

[7] Christoph W Borst and Arun P Indugula. 2006. A spring model for whole-hand virtual grasping. *Presence: Teleoperators & Virtual Environments* 15, 1 (2006), 47–61.

[8] Andreja Bubic, D Yves Von Cramon, and Ricarda I Schubotz. 2010. Prediction, cognition and the brain. *Frontiers in human neuroscience* 4 (2010), 25.

[9] Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. 2013. UltraHaptics: multi-point mid-air haptic feedback for touch surfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 505–514.

[10] Seungmoon Choi and Katherine J Kuchenbecker. 2012. Vibrotactile display: Perception, technology, and applications. *Proc. IEEE* 101, 9 (2012), 2093–2104.

[11] Andy Clark. 2013. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences* 36, 3 (2013), 181–204.

[12] Paul R Davidson and Daniel M Wolpert. 2005. Widespread access to predictive models in the motor system: a short review. *Journal of neural engineering* 2, 3 (2005), S313.

[13] Hannes Eisler and Chris Ottander. 1963. On the problem of hysteresis in psychophysics. *Journal of Experimental Psychology* 65, 6 (1963), 530.

[14] Tom Erez, Yuval Tassa, and Emanuel Todorov. 2015. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 4397–4404.

[15] Christer Ericson. 2004. *Real-time collision detection*. CRC Press.

[16] Marc O Ernst and Martin S Banks. 2002. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* 415, 6870 (2002), 429.

[17] James A Ferwerda. 2003. Three varieties of realism in computer graphics. In *Human Vision and Electronic Imaging VIII*, Vol. 5007. International Society for Optics and Photonics, 290–297.

[18] Scott Frees and G Drew Kessler. 2005. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.* IEEE, 99–106.

[19] Scott Frees, G Drew Kessler, and Edwin Kay. 2007. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)* 14, 1 (2007), 2.

[20] Karl Friston. 2010. The free-energy principle: a unified brain theory? *Nature reviews neuroscience* 11, 2 (2010), 127.

[21] Michael SA Graziano. 1999. Where is my arm? The relative role of vision and proprioception in the neuronal representation of limb position. *Proceedings of the National Academy of Sciences* 96, 18 (1999), 10418–10421.

[22] Takuto Hamano, Masahiko Onosato, and Fumiki Tanaka. 2016. Performance comparison of physics engines to accelerate house-collapsing simulations. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 358–363.

[23] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[24] Seongkook Heo, Christina Chung, Geehyuk Lee, and Daniel Wigdor. 2018. Thor's hammer: An ungrounded force feedback device utilizing propeller-induced propulsive force. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 525.

[25] Koichi Hirota and Michitaka Hirose. 2003. Dexterous object manipulation based on collision response. In *IEEE Virtual Reality, 2003. Proceedings.* IEEE, 232–239.

[26] Daniel Holz, Sebastian Ullrich, Marc Wolter, and Torsten Kuhlen. 2008. Multi-contact grasp interaction for virtual environments. *JVRB-Journal of Virtual Reality and Broadcasting* 5, 7 (2008).

[27] Johannes Hummel, Robin Wolff, Tobias Stein, Andreas Gerndt, and Torsten Kuhlen. 2012. An evaluation of open source physics engines for use in virtual reality assembly simulations. In *International Symposium on Visual Computing*. Springer, 346–357.

[28] Seungwoo Je, Myung Jin Kim, Woojin Lee, Byungjoo Lee, Xing-Dong Yang, Pedro Lopes, and Andrea Bianchi. 2019. Aero-plane: A Handheld Force-Feedback Device that Renders Weight Motion Illusion on a Virtual 2D Plane. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 763–775.

[29] Seungwoo Je, Hyelip Lee, Myung Jin Kim, and Andrea Bianchi. 2018. Wind-blaster: a wearable propeller-based prototype that provides ungrounded force-feedback. In *ACM SIGGRAPH 2018 Emerging Technologies*. ACM, 23.

[30] Kenneth O Johnson. 2001. The roles and functions of cutaneous mechanoreceptors. *Current opinion in neurobiology* 11, 4 (2001), 455–461.

[31] Shunichi Kasahara, Jun Nishida, and Pedro Lopes. 2019. Preemptive Action: Accelerating Human Reaction using Electrical Muscle Stimulation Without Compromising Agency. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 643.

[32] Alexander Kulik. 2009. Building on realism and magic for designing 3D interaction techniques. *IEEE Computer Graphics and Applications* 29, 6 (2009), 22–33.

[33] Byungjoo Lee, Qiao Deng, Eve Hoggan, and Antti Oulasvirta. 2017. Boxer: a multimodal collision technique for virtual objects. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ACM, 252–260.

[34] Byungjoo Lee, Sunjun Kim, Antti Oulasvirta, Jong-In Lee, and Eunji Park. 2018. Moving target selection: A cue integration model. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 230.

[35] Jungjin Lee, Bumki Kim, Kyehyun Kim, Younghui Kim, and Junyong Noh. 2016. Rich360: optimized spherical representation from structured panoramic camera arrays. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 63.

[36] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology* 49, 4 (2013), 764–766.

[37] Pedro Lopes, Alexandra Ion, and Patrick Baudisch. 2015. Impacto: Simulating physical impact by combining tactile stimulation with electrical muscle stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 11–19.

[38] Pedro Lopes, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. 2017. Providing haptics to walls & heavy objects in virtual reality by means of electrical muscle stimulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 1471–1482.

[39] Ronald G Marteniuk and Martha L Ryan. 1972. Psychophysics of kinesthesis: Angular movement. *Journal of motor behavior* 4, 3 (1972), 135–142.

[40] Madjid Mashhour. 1964. Psychophysical relations in the perception of velocity. (1964).

[41] Daniel Mendes, Fabio Marco Caputo, Andrea Giachetti, Alfredo Ferreira, and J Jorge. 2019. A survey on 3d virtual object manipulation: From the desktop to immersive virtual environments. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 21–45.

[42] Daniel Mendes, Filipe Relvas, Alfredo Ferreira, and Joaquim Jorge. 2016. The benefits of dof separation in mid-air 3d object manipulation. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. ACM, 261–268.

[43] Farouk Messaoudi, Gwendal Simon, and Adlen Ksentini. 2015. Dissecting games engines: The case of Unity3D. In *2015 International Workshop on Network and Systems Support for Games (NetGames)*. IEEE, 1–6.

[44] Mark R Mine, Frederick P Brooks Jr, and Carlo H Sequin. 1997. Moving objects in space: exploiting proprioception in virtual-environment interaction.. In *SIGGRAPH*, Vol. 97. 19–26.

[45] Prerna Mishra and Urmila Shrawankar. 2016. Comparison between Famous Game Engines and Eminent Games. *International Journal of Interactive Multimedia & Artificial Intelligence* 4, 1 (2016).

[46] Sean C Mondesire, Douglas B Maxwell Jonathan Stevens, Steven Zielinski, and Glenn A Martin. 2016. Physics Engine Benchmarking in Three-Dimensional Virtual World Simulation. *MODSIM World* (2016), 5–8.

[47] Matthew Moore and Jane Wilhelms. 1988. Collision detection and response for computer animation. In *ACM Siggraph Computer Graphics*, Vol. 22. ACM, 289–298.

[48] Thi Thuong Huyen Nguyen, Thierry Duval, and Charles Pontonnier. 2014. A new direct manipulation technique for immersive 3d virtual environments. In *ICAT-EGVE 2014: the 24th International Conference on Artificial Reality and Telexistence and the 19th Eurographics Symposium on Virtual Environments*. 8.

[49] Carol O'Sullivan. 2005. Collisions and attention. *ACM Transactions on Applied Perception (TAP)* 2, 3 (2005), 309–321.

[50] Carol O'Sullivan and John Dingliana. 2001. Collisions and perception. *ACM Transactions on Graphics (TOG)* 20, 3 (2001), 151–168.

[51] Carol O'Sullivan, John Dingliana, Thanh Giang, and Mary K Kaiser. 2003. Evaluating the visual fidelity of physically based animations. In *ACM Transactions on Graphics (TOG)*, Vol. 22. ACM, 527–536.

[52] Carol O'Sullivan, Sarah Howlett, Yann Morvan, Rachel McDonnell, and Keith O'Conor. 2004. Perceptually adaptive graphics. *Eurographics state of the art reports* 4 (2004), 1–24.

[53] Carol O'Sullivan, Ralph Radach, and Steven Collins. 1999. A model of collision perception for real-time animation. In *Computer Animation and Simulation'99*. Springer, 67–76.

[54] Antti Oulasvirta, Sunjun Kim, and Byungjoo Lee. 2018. Neuromechanics of a button press. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 508.

[55] Jong-Rack Park, Dae-Woon Lim, Soo-Yeol Lee, Hang-Woon Lee, Mi-Hyun Choi, and Soo-Cheol Chung. 2008. Long-term study of simulator sickness: differences in EEG response due to individual sensitivity. *International Journal of Neuroscience* 118, 6 (2008), 857–865.

[56] Roger Ratcliff. 1978. A theory of memory retrieval. *Psychological review* 85, 2 (1978), 59.

[57] Roger Ratcliff and Gail McKoon. 2008. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation* 20, 4 (2008), 873–922.

[58] Paul SA Reitsma and Carol O'Sullivan. 2009. Effect of scenario on perceptual sensitivity to errors in animation. *ACM Transactions on Applied Perception (TAP)* 6, 3 (2009), 15.

[59] Steven Robbins, Edward Waked, and Jacqueline McClaran. 1995. Proprioception and stability: foot position awareness as a function of age and footware. *Age and Ageing* 24, 1 (1995), 67–72.

[60] Paul G Ronco. 1963. An experimental quantification of kinesthetic sensation: Extent of arm movement. *The Journal of Psychology* 55, 1 (1963), 227–238.

[61] Arne Rönnau, F Sutter, Georg Heppner, Jan Oberländer, and Rüdiger Dillmann. 2013. Evaluation of physics engines for robotic simulations with a special focus on the dynamics of walking robots. In *2013 16th International Conference on Advanced Robotics (ICAR)*. IEEE, 1–7.

[62] Fabrice R Sarlegna and Robert L Sainburg. 2009. The roles of vision and proprioception in the planning of reaching movements. In *Progress in Motor Control*. Springer, 317–335.

[63] Rajinder Sodhi, Ivan Poupyrev, Matthew Glisson, and Ali Israr. 2013. AIREAL: interactive tactile experiences in free air. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 134.

[64] Ian Spence. 1990. Visual psychophysics of simple graphical elements. *Journal of Experimental Psychology: Human Perception and Performance* 16, 4 (1990), 683.

[65] Anthony Steed. 2006. Towards a general model for selection in virtual environments. In *3D User Interfaces (3DUI'06)*. IEEE, 103–110.

[66] Frank Steinicke and Gerd Bruder. 2014. A self-experimentation report about long-term use of fully-immersive technology. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. ACM, 66–69.

[67] Stanley S Stevens. 1957. On the psychophysical law. *Psychological review* 64, 3 (1957), 153.

[68] Yuriko Suzuki and Minoru Kobayashi. 2005. Air jet driven force feedback in virtual reality. *IEEE computer graphics and applications* 25, 1 (2005), 44–47.

[69] John C Tuthill and Eiman Azim. 2018. Proprioception. *Current Biology* 28, 5 (2018), R194–R203.