

Laplacian Vision: Augmenting Motion Prediction via Optical See-Through Head-Mounted Displays

Yuta Itoh¹ Jason Orlosky² Kiyoshi Kiyokawa³ Gudrun Klinker⁴

^{1,4}Technical University of Munich, Garching, Germany
{itoh,klinker}@in.tum.de

^{2,3}Osaka University, Osaka, Japan
{orlosky@lab., kiyoo@} ime.cmc.osaka-u.ac.jp

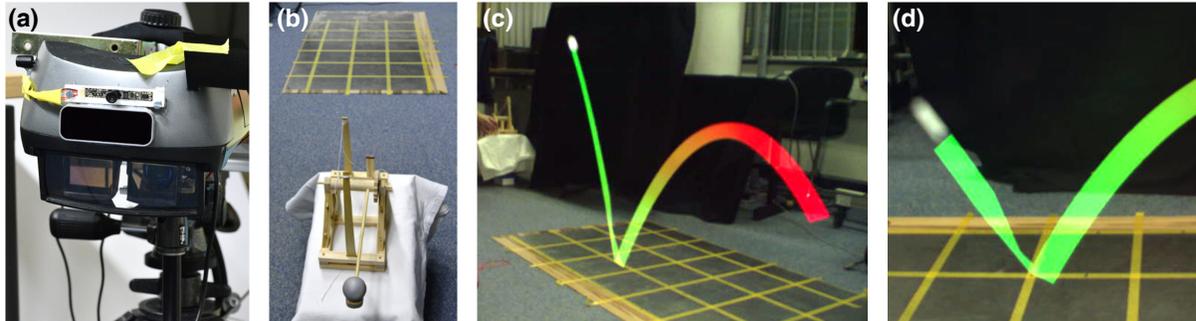


Figure 1: (a) Our Laplacian Vision system with an Optical See-Through Head-Mounted Display (OST-HMD) and user-view camera placed *behind* the display screen. (b) A miniature catapult launches a target ball to a landing pad. (c) A scene with see-through visualization of a predicted ball trajectory, which is superimposed onto the view in real-time. (d) A magnified view through the system of the landing area from the side.

ABSTRACT

Naïve physics [7], or folk physics, is our ability to understand physical phenomena. We regularly use this ability in life to avoid collisions in traffic, follow a tennis ball and time the return shot, or while working in dynamic industrial settings. Though this skill improves with practice, it is still imperfect, which leads to mistakes and misjudgments for time intensive tasks. People still often miss a tennis shot, which might cause them to lose the match, or fail to avoid a car or pedestrian, which can lead to injury or even death.

As a step towards reducing these errors in human judgement, we present Laplacian Vision (LV), a vision augmentation system which assists the human ability to predict future trajectory information. By tracking real world objects and estimating their trajectories, we can improve a user's prediction of the landing spot of a ball or the path of an oncoming car. We have designed a system that can track a flying ball in real time, predict its future trajectory, and visualize it in the user's field of view. The system is also calibrated to account for end-to-end delays so that the trajectory appears to emanate forward from the moving object. We also conduct a user study where 29 subjects predict an object's landing spot, and show that prediction accuracy improves 3 fold using LV.

CCS Concepts

- Human-centered computing → Visualization theory, concepts and paradigms; *Information visualization*;
- Applied computing → *Physics*;

Keywords

Vision augmentation, prediction, tracking, augmented reality, HMD

1. INTRODUCTION

Naïve physics refers to the human skill involved in understanding physical phenomena [7], and is critical for daily activities. It is used to avoid collisions in traffic, play any kind of timing intensive sport, and conduct the most basic motor tasks in various settings. However, this ability is limited in many ways, and real world physics is often complex and hard to predict. For example, predicting a final landing spot or intersection of a flying ball can be a difficult task for us if there are strong winds, the ground is uneven, or another player is occluding a clear view of the ball. Predicting the trajectory of an oncoming car or hazardous object can be difficult if you are looking in the other direction, have limited depth perception, or have little time to make the prediction.

On the other hand, computers can often exceed humans in their abilities to predict information, especially trajectories that involve more complex (but constant) physical properties. When humans make these mental calculations, compounding of errors in the estimation of bounce or ricochet trajectories can result in very large errors in estimation of the final landing spot, whereas a machine can more accurately predict the final landing spot of a ball after bouncing off a number of walls by making an estimation using calculations based on Newtonian physics.

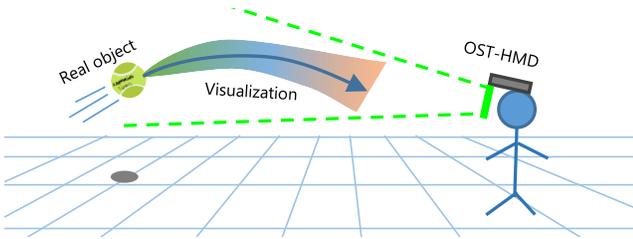


Figure 2: The concept behind our Laplacian Vision system. By integrating predictive information from an object in motion into a user’s field of view, the user will be able to enhance his or her perception of its future path using the information presented by the HMD.

We think that using a machine’s ability to make these calculations quickly can significantly augment a human’s ability to improve estimation accuracy, make sound judgments, and ultimately avoid accidents. We can possibly even enable humans to *foresee* a short-term future of the world – as if we acquire the ability of the *Laplace’s demon*, a thought experiment in the causal determinism in canonical physics articulated by a physicist Pierre-Simon Laplace in the 19th century [12].

As a step towards this goal, we present Laplacian Vision, a vision augmentation system designed to assist naïve physics (Fig. 1, 2). Using an optical see-through head mounted display and object sensing/tracking, our system predicts and overlays the trajectory of a moving object onto the real world from the user’s perspective. The system consists of four main parts, including a 1) prototype optical see-through head mounted display (OST-HMD), 2) external IR cameras to facilitate tracking of an airborne object, 3) the trajectory prediction and game engine integration, and 4) a calibration step to correctly align the augmentation with a user’s eye.

1.1 Contributions

In addition to building the system itself, our main contributions include:

- demonstrating the proof-of-concept system which displays an airborne ball and its future trajectory from a first-person perspective with a user-perspective camera in real time
- conducting an experiment that evaluates the improvements in trajectory estimation using Laplacian Vision versus unassisted human vision
- investigating technical challenges in realizing naïve physics augmentation and defining new areas for exploration

2. RELATED WORK

Our system draws from ideas in a number of different fields, including object tracking and prediction, vision science and augmentation, and spatial augmented reality (AR).

2.1 Tracking of Moving Objects and Spatial Augmented Reality

For most AR applications, some combination of wearable or external tracking systems are necessary. This is even more essential for predictive systems, which need both speed

and accuracy to assure that predictive information is correct. For example, if we want to predict the trajectory of a moving ball in someone’s field of view, we need a system that can both track the 3D position of the ball and constantly update the ball’s trajectory model for prediction at the same time. Tracking of objects in motion is a relatively mature topic in computer vision, including high-speed vision systems that have been applied to ping-pong ball trajectories (100 fps) [24], juggling balls (250 fps) [11], rain drops (500 fps) [3], and snow flakes (1,000 fps) [21]. Interestingly, these systems also employ kinematic prediction models to ensure stable tracking, excluding the last one, which frame rate is high enough to allow perceptual consistency. The tracking system closest to what we have implemented is by Koike and Yamaguchi [11], who developed a system for tracking and augmenting small balls with a projector. A very similar work by Okumura et al. improved further on this type of high-speed projection by integrating galvanometer mirror optics [16].

However, most of the aforementioned systems are designed simply to project images on the tracked objects for basic augmentations, and do not consider visualizing future trajectories. There have been several such works that do employ prediction for the purpose of controlling physical systems. One prime example is that of Zhang et al., who used predicted trajectory information to control a robotic arm and return incoming ping pong balls fed from a human [24]. This is conceptually similar to our design in some ways, but we need to overlay information to a human user through an HMD, calibrate the system so that the AR is correctly perceived, and adjust prediction to compensate for tracking and rendering delay.

Note that while our system requires tracking for correct prediction, the tracking algorithm itself is not our primary contribution. Low latency tracking has also been applied to programmable headlights by Tamburo et al., which were designed to sense and adapt to their surroundings to improve safety for night driving [21]. This idea was taken a step further by Charette et al., who used a projector based headlight to avoid illuminating individual raindrops or snowflakes to improve driver vision during hazardous driving conditions [3]. Rays of light only passed through dark spots in the user’s field of view, leaving droplets, snowflakes, or hail stones unilluminated for a clearer view of the road.

Trajectory prediction has been applied in some other interesting ways, such as interactive paper airplane design. Umetani et al. built a system called Pteromys that could visualize the approximate trajectory of a paper airplane based on the airplane’s shape and construction [23]. Though this system was designed for offline use, it demonstrates how trajectory information can be used to enhance creative experiences for users.

2.2 Vision Science and Augmentation

Research has shown that motion estimation and hand-eye coordination are complex tasks, involving a number of different processes including vestibulo-ocular eye movements, vergence, smooth-pursuit, saccades and accommodation [2].

This kind of research is essential when designing vision augmentation systems since augmentations typically assist or utilize one or more of these visual functions. For example, a system called SpiderVision proposed by Fan et al. allowed users to see images from a camera mounted to the rear of

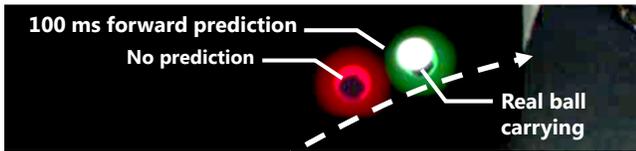


Figure 3: Visualization of latency compensation. In our system, rendering a halo at the latest position of a flying ball takes roughly 100 ms (4 frames in 50 fps) which causes misalignment against the real world (the red halo). Compensating the delay by prediction corrects this misalignment (the green halo).

an HMD when moving objects passed by [6].

Other types of enhancement systems including those that improve readability or visibility, such as that by Itoh et al. They utilized a compensation filter to eliminate visual aberrations and enhance images of characters for improving readability [9]. Another system similar to this by Orlosky et al. proposed the concept of modular camera-lens based augmentations, which allow users to quickly switch between augmentations such as binocular zoom or field of view expansion [18]. This system also utilized eye tracking to allow for eye-based control of desired augmentations.

The eyes are also very closely tied to the continuous prediction of motion [4], which supports our hypothesis that overlaying virtual information can improve or enhance the prediction of future trajectories or landing spots.

2.3 Low Latency Rendering

Even if we perfectly predicted a short-term future trajectory of a target object, it would be meaningless if the system latency caused virtual information to appear behind the ball (as shown on Fig. 3 left as an example of such a case). To ensure that predictive information is actually useful, the system must complete the tracking, prediction, and rendering processes within a reasonable time interval so that the displayed trajectory information appears in front of the moving object. Removing this delay is essentially impossible as the virtual world always lags behind the real world, i.e., first we have to wait for the motion in the real world before we can decide what to render.

Considerable work has been conducted that investigates how much rendering delay is noticeable to humans in a virtual reality environment, which suggests that human latency discrimination spans are less than 15 ms [14] or 16 ms [5]. This type of work has also been applied to visual-physical systems such as touch based interfaces. In a spatial AR context, which is relatively similar to our setup, Joda et al. found that the minimum latency people would notice could be as low as 2.38 ms when dragging virtual objects on a real surface [10].

If a system achieves end-to-end latency below these thresholds, it becomes, effectively, a no-delay system. In general, however, displays including OST-HMDs have bottlenecks in rendering delays and their fixed display update time, e.g., 33.3 ms for 30-fps displays. A GPU renders a frame while the previous frame is being sent out to a display panel (known as scanout). To render a virtual trajectory, the GPU has to fetch the trajectory data at the time rendering started. This inevitably inserts at least one frame delay between the real and the virtual world.

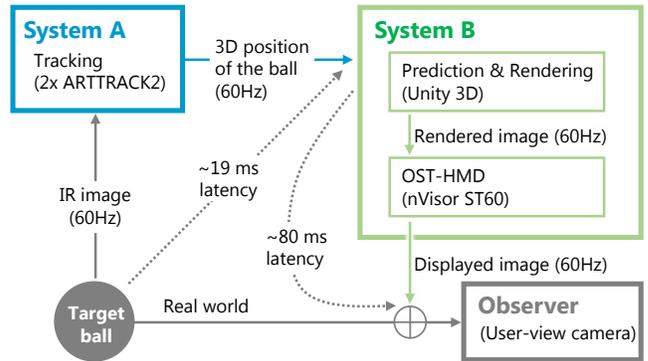


Figure 4: Block diagram of data flow in our system. To visualize future motion without latency to the user, we have to minimize the latency from systems A and B and/or improve the prediction accuracy at System A. Each block represents a different physical machine.

One option is to apply a post-rendering technique, such as that of Smit et al. or Mark et al. [20, 15]. This kind of technique adjusts the rendered image in 2D transformation based on the last-millisecond tracking data of the target.

We have been able to implement a somewhat different strategy for solving this problem since we are tracking objects that have a high kinetic component. One good thing about most moving objects is that they are bound by Newtonian Physics, i.e., we can deterministically calculate its positions in time by the control theory [11] (Sec. 3.5). Therefore, it is possible to predict almost exactly where the ball will be after all processing has occurred. Essentially, we include the frame delay in the trajectory prediction model – which allows us to render the virtual trajectory ahead of time in our proof-of-concept system.

3. HARDWARE SETUP AND CALIBRATION

In short, our goal was to develop a system which can provide a user with future trajectory data by predicting and overlaying that information through an OST-HMD.

To realize this goal, our system needed to

- predict, in real time, the motion of a physical object in the scene
- render a virtual representation of the object’s trajectory on the display so that its appearance is spatially consistent with the object.
- guarantee that the end-to-end latency from prediction to rendering is small enough for a given task

Each of these requirements has its own implementation challenges, but the real obstacle was trying to get all of these parts working at the same time on a rather restricted hardware setup.

We first describe the data flow of the system as well as assumptions that we make for our test implementation. We then elaborate on the design of the hardware and software frameworks we developed to implement Laplacian Vision.

3.1 System Dataflow

Figure 4 illustrates the general flow of data in our system. First of all, we throw a retroreflective target ball into the

scene. The tracking system (System A) captures the target ball using two infrared cameras and then computes the current position of the ball. The rendering/display system (System B) then receives this calculated 3D position as UDP packets via local Ethernet connection at 60Hz. The data is first received by the Unity 3D game engine, where we conduct prediction (Sec. 3.5) and rendering. The resulting virtual information is rendered at 60Hz and image frames are sent to the OST-HMD. Finally, a user-perspective camera (Observer) simultaneously captures both the tracked ball in the real world and overlaid images rendered on the display.

As seen from Fig. 4, the system inevitably holds a certain latency due to the tracking, prediction, and rendering processing. Naive rendering would leave the rendered trajectory behind the actual ball position. For instance, Fig. 3 (left) shows a captured image from the user-perspective camera with a target ball flying from right to left with a virtual red halo. Even though the halo was rendered based on the measured 3D position given from the tracker in real time, the halo was rendered at the position where the ball was at roughly 100 ms prior, which would be the end-to-end latency of the system in a naive rendering scheme. Note that the System B seems to have approximately 80 ms latency as a result since the tracking system has 19 ms latency.

3.2 Assumptions

Since our system focuses on proving the LV concept on an OST-HMD system, we are not focused on the tracking algorithms with prediction or the low-latency rendering as a contribution. We use an industrial outside-in tracking system (Sec. 3.3) and its spherical tracking target as a physical target object of which we predict a trajectory. We also assume constant system latency based on our heuristic measurement.

It is important to note that our prototype does not yet incorporate head tracking, and still requires tests with user predictions in real-world, dynamic situations. For the proof of concept, we only test with a perspective camera to emulate static viewpoints of a user (Sec. 3.4). Additional calibration and perspective calculations would be required for a moving viewpoint.

We also limit the motion of the target ball by simply catapulting it into mid air in a roughly repeatable trajectory in an indoor environment. This reduces the complexity of the prediction problem to allow for clearer demonstration and evaluation of the system (Sec. 3.5).

3.3 Hardware Layout

Figure 5 shows our proof-of-concept system setup. We used an nVisor ST60 OST-HMD from NVIS, with a 1280 x 1024 resolution and 60° diagonal field of view. As a real world object, we used a retroreflective sphere with a diameter of 40 mm. To track the sphere, we employed an external tracking system consisting of two ARTTRACK2 cameras hung on the ceiling of the room. This system provides the 3D position of the sphere at 60Hz, and tracking accuracy is less than one millimeter as stated by the manufacturer.

For demonstration purposes, we installed a user-perspective camera placed behind the left optical element of the display so that the camera captures the same scene that would be presented to the user, though the image is not stereoscopic. As the user-perspective camera, we use IDS UI-1240ML-C-HQ with a 4-mm C-mount lens. By tuning the camera’s

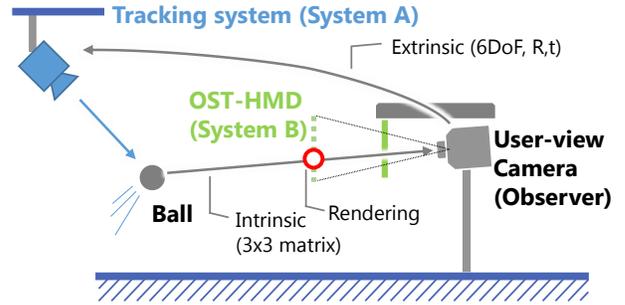


Figure 5: Diagram of our hardware setup from the side, including calibration parameters necessary for correct rendering. The extrinsic parameter includes rotation (R) and translation (t) from the tracking system to the eye-HMD coordinate system.

region of interest, we get a 632×492 , 50Hz image feed.

3.4 System Calibration

Overlaying virtual content on the real scene requires calibration of the OST-HMD with respect to the current viewpoint of a user, in our case the user-perspective camera. In other words, we have to define a virtual camera so that a virtual content rendered by a graphics pipeline overlaps with the corresponding physical object [22, 1]. We calibrate our display system based on a manual method called Single Point Active Alignment Method (SPAAM) [22].

Under SPAAM, the eye-HMD system, which consists of an eyeball center and the display screen, is treated as an off-axis pinhole camera. The camera model consists of extrinsic parameters, which consist of a 6DoF transformation from the world coordinate system to the eye-HMD coordinate system with the eyeball center as the origin and the display angle as orientation axes, and an intrinsic parameter, which is a 3×3 camera matrix of the eye-HMD camera (Fig. 5). Concatenating both parameters result in a 3×4 projection matrix. The spatial calibration of OST-HMDs is to estimate this matrix.

To estimate the projection matrix in SPAAM, the user needs to align a 3D world reference to a 2D point shown on the display. This procedure gives a pair of 3D-2D correspondences. Since the projection matrix contains 11 variables with unknown scale, this procedure is repeated in 3D with different depths and 2D points at least 6 times to estimate the matrix. In our setup, we used a target ball with 2cm diameter as a 3D reference target, and we align the marker to a crosshair rendered in a random position on the display through the view from the user-perspective camera.

3.5 Motion Prediction and Visualization

Similar to [11], we employ a Kalman filter with a linear kinematic model with the gravity vector in the motion prediction. Each time the display system (System B) receives the 3D positions of the ball from the tracking system (System A), the display system updates the current position and velocity of the ball.

We update the latest position and velocity every time we receive a new ball position from the tracker, and render a trajectory calculated from the motion model at each display frame. We also measured the ground level of the scene beforehand, and use that information to make the trajectory

bounce off of a virtual plane aligned with the physical ground in the real world. All of the virtual objects and visualizations were created using Unity 3D, version 5.2. To compensate for the end-to-end latency (about 100 ms) when rendering the trajectory, we increment the trajectory’s path out several frames so that it appears in front of the ball, as can be seen in Figure 6. In other words, the trajectory will always be ahead of the current measured position using this prediction, which effectively makes the trajectory appear as if it is a zero-delay system.

4. EVALUATION

To evaluate how well our LV system can assist users in basic prediction tasks, we designed a user study where 29 subjects were asked to predict the location of impact of a moving object. To do so, subjects watched a series of user-perspective *videos* of a ball flying across a scene.

We used recorded videos to ensure consistent viewing and fair estimation across all subjects. Although the videos lack stereo information, we have found that the current evaluation already provides considerable evidence that LV augmentation significantly benefits naïve physics judgments.

Each video was presented both with and without the LV system, and subjects were instructed to use a cursor to select the point at which they thought the ball would impact the floor. More specific examples of these scenes are shown in Fig. 6, with the top row representing with and without LV conditions. The middle row shows conditions where the videos were stopped and looped prior to impact, and the last two show different perspective angles. Reasoning behind these conditions is as follows.

The purpose of our experiments is to prove our LV system can improve predictions in a very basic use case. For example, the system would be impractical if users would not even benefit from trajectory information provided from a static viewpoint. We first wanted to test simple judgments when the ball is coming from a particular direction, so we designed a *View Angle* condition (Sec. 4.1.1). Moreover, the system would likely be useless if it only helped users after they watched the ball actually bouncing on the ground, so we also designed a *Video Length* condition (Sec. 4.1.2).

4.1 Experimental Setup

To record the scenes with the ball flying through the air, we launched it using a miniature catapult (Thumbs Up Da Vinci Catapult, Fig. 1 (b)) so that we could create controlled paths over the different conditions. To analyze impact points, we placed a metal plate on the ground, onto which we taped a grid with 20 centimeter spacing. The plate was heavy enough so that the impact did not move it. We then let the user-perspective camera record various scenes with/without the LV visualization for three *View Angle* conditions.

Video Length was adjusted to be consistent over all videos. Recording was started at the instant the ball was launched until shortly after (3 frames) the first impact. The reason we recorded few buffer frames after the impact is because it is hard to measure the impact time without checking frames after which the ball bounced. As a result, videos with the catapult in the scene were 1 ± 0.02 sec (from 49 to 51 frames in 50 fps).

Next, we elaborate the experiment conditions we introduced to compare the LV system with unassisted vision.

Estimation game base conditions:

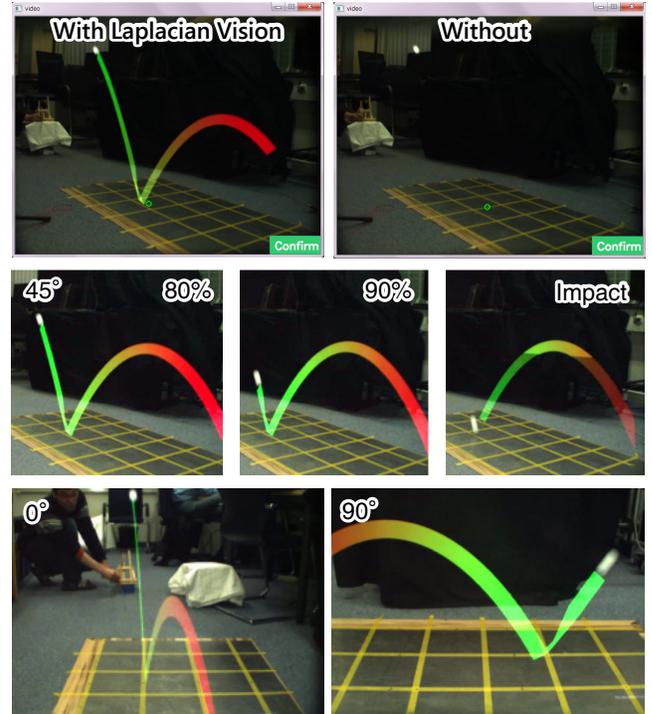


Figure 6: Representative user study conditions. Subjects watched 84 different videos of ball-launch scenes taken from the user-perspective camera. The top row shows frames from our estimation game both with and without LV. The small green circles represent the estimated impact point of a user for each video. The middle row shows three *Video Length* conditions, 80%, 90%, and 100% (the frame of an actual ball impact). The last row shows two additional viewing angles we tested, which were also combined with the different video lengths, though not shown here. We thus have three different *View Angle* conditions of roughly 45°, 0°, and 90°.

4.1.1 View Angle condition

The first condition is the viewing perspective of the user-perspective camera. We hypothesized that different perspectives of the ball path toward the subjects would affect their abilities to judge impact, so we placed the HMD-camera system in three different viewing positions. First, the system faces the catapult over the impact area. This makes the angle between the line of fire of the catapult and that of sight of the camera roughly 0°. We call this angle *View Angle*, which is 0° for this case. The system was also oriented so that the angle was roughly 45°, and finally from the side (90°). Fig. 6 left column shows the example of user-perspective images of these three conditions.

4.1.2 Video Length condition

The second condition is the frame at which we stop the video and loop. This was designed to test subjects’ abilities to determine the future trajectory of the ball earlier on in its flight path. This *Video Length* condition included 80%, 90%, and 100% (few frames after impact point) playbacks, which let us test estimation abilities with different amounts

of limited information. The last frames in each of these videos can be seen in the middle row of Fig. 6 .

For each *View Angle* condition, we launched the ball 5 times. Thus, we recorded 30 launch videos in total,

$$5 \times LV\{\text{on, off}\} \times \text{ViewAng.}\{0^\circ, 45^\circ, 90^\circ\} = 30.$$

Now, by taking the *Video Length* condition into account, we have impact prediction sessions per user,

$$5 \times LV \times \text{ViewAng.} \times \text{VideoLen.}\{80\%, 90\%, 100\%\} = 90.$$

Note that we excluded two videos from the recordings: one from the (off, 45°) condition and the other from the (off, 90°), resulting in 84 total sessions per subject. This was because the impact points in the videos were either very close to the edge of the image or out of the image.

Though we initially recorded a condition with 100% play time, i.e. subjects saw actual impacts of the ball regardless of the LV visualization, we disregard this condition from the main analysis. Because there was almost no significant difference in the prediction error (defined in Sec. 4.3) in whether showing the visualization or not (Sec. 4.3.2). We find this is reasonable since subjects can simply watch the actual impact and ignore any predictive information. The 100% condition is still mentioned briefly in the discussion section since we found that the visualization actually had a minor influence for this condition.

4.2 Data Collection

We let each subject play the game as shown in shown in Figure 7. Each subject watched the above 56 sessions (videos) in randomized order. For each session, the corresponding video looped endlessly until the subject finalized his or her choice of the expected impact point. To input the point, subjects could click any point in the video using a mouse. After clicking, a green ring was rendered at the click point with the video still looping in the background. The subject could click another point to change his or her choice, and finally clicked a confirm button to record the choice and move to the next session. The window size of the game was fixed to the original size of the videos, and the subjects were not allowed to change the image size during the sessions.

We distributed the game to subjects electronically, and the subjects played the game on their own personal monitors. We collected data of 29 subjects with 24 males and 5 females. The average and median age of the subjects are 28.6 and 25, respectively.

4.3 Error Analysis

We first explain how we measured the ground truth for impact points. We checked all videos on a frame-by-frame basis and identified frames before and after the impact in each video. Based on the two frames, we marked a ground point at which the ball would have touched at the time of the impact. We treat these manually measured points as the ground truth points. At the same time, we also marked each grid points in the images. We use these grid points to convert our measurements from 2D image space to points in the 3D, real scale coordinate system on the metal plate.

Since we know that the squares on the grid have an edge-length of 20 centimeters, we compute a 2D homography to the real grids from the 2D points we marked. By applying the homography matrices, we can convert all measurements into a bird’s-eye view together with the image region of the ground (Fig. 7, Sec. 4.3.1). For each video, we then converted the ground truth points and the users’ input points in

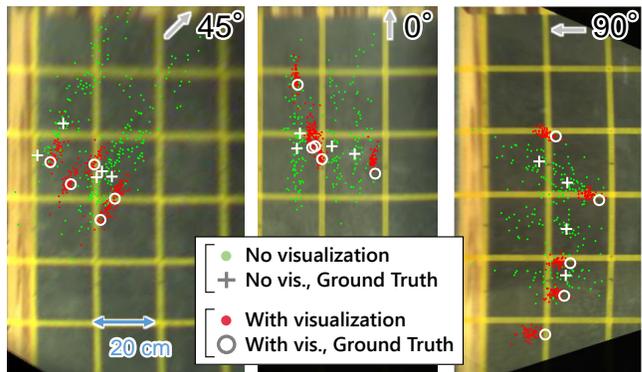


Figure 7: Visualization of user input for each *View Angle* condition from a bird’s-eye view. Green dots are inputs over the all subjects in sessions without LV and red are with LV. From left to right: 45° , 0° , and 90° . The results show that LV visualization reduces the variance of point estimation among subjects. The *Video Length* condition is mixed.

the uv space to xy points on the board. The error measurement is then defined as the real-scale distance between an input and a corresponding ground truth point. Note that the homography matrices from the same *View Angle* condition are almost identical, thus we merged all points according to their *View Angle* condition in the bird’s-eye view plots.

After computing the errors for all subjects, we applied an analysis of variance (ANOVA) over the three conditions: the use of LV visualization, *View Angle*, and *Video Length*. In short, our analysis suggests that the LV visualization significantly improves the prediction accuracy of subjects, excluding the 100% *Video Length* case (Sec. 4.3.2).

4.3.1 Qualitative Analysis: Bird’s-eye View Data

The bird’s-eye view in Fig. 7 gives us an intuitive visualization of all user placements overlaid onto a single image. In the figure, red represents users’ choices with LV assistance and green with no assistance. The user placements are only from the *Video Length* condition of 80% and 90%, and merged together in the figure.

One interesting observation is that we can clearly see LV creates distinctive clusters of impact points over user inputs. The deviation of the baseline plots, i.e., without our visualization, shows that subjects are less certain about the ball trajectory.

On the other hand, these deviations have certain directional tendencies, which are different from each viewing angle. We speculate this is due to the slightly misaligned visualization and/or difficulty with guessing depth information in the current user study design. Still, these results suggest that our method significantly reduces error variances even without a stereoscopic view, especially given the fact that the system even works at a near 90° perspective – a viewpoint that does not require strong depth perception for prediction. Developing further evaluations of varied trajectories of different objects in more complex environments will make for interesting future work.

4.3.2 Quantitative Analysis: Error Statistics

Figure 8 shows boxplots of the errors over each of the conditions. We used repeated-measures ANOVA by treating

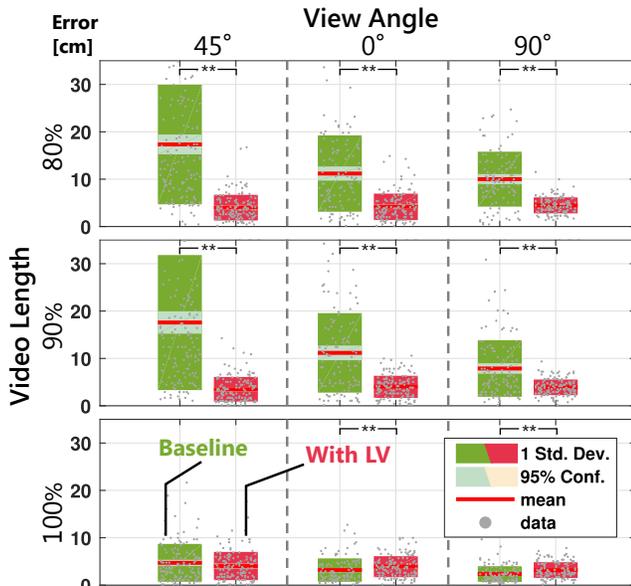


Figure 8: Plot showing error for the *View Angle* (x axis) and *Video Length* (y axis) conditions. Green represents error without visualization and red represents error using LV. -** indicates that the corresponding pair has a significant difference.

videos for each condition combinations as the random variable. Note that, after observing the Q-Q plot of the original errors for LV conditions, we decided to use the square root of the errors in the analysis to improve the data normality.

First of all, Mauchly’s test indicated non-sphericity on the measurement with $\epsilon = \{0.135, 0.238\}$ for the Greenhouse-Geisser and the Huynh-Feldt approximation, respectively. By the ANOVA test, we found a significant main effect that showing the LV visualization significantly reduces the error compared to no-visualization ($F_{1,27} = 42.28, p \ll 0.01$). We found an interaction effect between the *View Angle* and *Video Length* conditions ($F_{4,108} = 5.13, p < 0.01$). Subsequently, we analyzed each pair of the two conditions and the LV condition with a post-hoc Tukey-Kramer honest significant difference test.

The test found that showing the LV visualization significantly reduces the error compared to no-visualization when the *Video Length* condition is either 80% or 90% in all the *View Angle* conditions with all $p \ll 0.01$. On the other hand, in the condition $\{0^\circ, 90^\circ\} \times 100\%$, viewing with the LV system significantly increased the error. This seems to be due to the fact that the visualization covered the real ball at the impact time and affected subjects’ judgments. In practice, however, predictive information would not be necessary if a user has already seen 100% of the ball’s trajectory.

We conclude that the LV system showed clear potential for assisting users’ naïve physics skills, and improved their visual judgment in predicting the behavior of a real object.

4.4 Discussion

Having shown that the LV system can assist our motion prediction ability, there are a number of interesting new questions to explore with this type of augmentation.

Visualization of prediction: It will be interesting to explore what kind of visualizations are suitable for other

applications with the LV system for assisting various predictions. Since our LV system visualizes information into a user’s egocentric vision directly via an OST-HMD, this type of first-person visualization can be applied to a number of industrial and commercial scenarios, especially with regards to safety. Such augmentations have been a key topic in AR for manufacturing [17] and logistics [19]. We hope that this work will encourage testing in environments where reducing collisions, accidents, and mistakes are critical in nature.

Sensorimotor evaluation: Although our LV system proved to assist a user’s *sensing*, further testing is needed for the user’s actions. I.e., it is still unclear how well our system can actually improve user’s motions or reaction times in a real world task. For example, it would be difficult for users to dodge an arrow, though chances of catching a flying ball or avoiding an oncoming object can likely be improved.

Therefore, investigating LV’s benefits in other sensorimotor tasks will likely establish it as a practical application.

Evaluation scheme for LV systems: Improving the evaluation scheme is also an important point to consider. In general, a proper evaluation of an LV system is very difficult due to its real time nature. For example, we can not freeze a physical flying ball before the impact and ask subjects to record their estimations. We will need to test actual reactions in follow-up studies.

From a perceptual point of view, it would also be valuable to build a stereo system to see how depth perception of users affects the effectiveness of the LV system, which might have caused the larger error variance in 45° setups.

5. FUTURE WORK

This work is our first trial to establish a practical Laplacian Vision system, and we believe the concept triggers many exciting research questions including the following.

Firstly, we can expand the prediction capability of a LV system by combining more sophisticated physics such as aerodynamics, kinematics like restitution, and non-rigid object models. For example, a tennis player can perfectly see the entire trajectory of a served ball from the opponent, bouncing points on the ground, and a swing point where the player has to swing through. Note that user’s action can also change the prediction state of an object [13].

Although we assumed the system predicts the real world, it can also render a future state of other intelligent systems such as a vehicle. These systems might broadcast expected future states of their own, e.g., a driving route. Our LV system then can integrate that information in the visualization. This, as a positive side effect, would also reduce the cost of prediction done by the LV system itself. Perhaps, the Internet of Things on the rise will serve as an infrastructure to realize a versatile LV system in the future.

Visualization is also a key topic to improve the performance of LV systems as we stated in Sec. 4.4. It would be very dependent on allowed time users can spend until they have to make a judgment for an action.

Last, but not least, developing a more practical LV system is beneficial to investigate the maximal potential such a system could provide to people in daily life. The first necessary step would be to integrate head tracking, and to test it with a real human viewer instead of a user-perspective camera. We would even need eye tracking for active exercises where a display could constantly moves on a wearer’s head [8].

6. CONCLUSION

We introduce Laplacian Vision (LV), a system for augmenting human vision with predictive information in real time. We then demonstrate the capabilities of the system by visualizing the future trajectory of a ball flying through the air. To do so, we combine object tracking, trajectory prediction, spatial calibration, delay compensation, and rendering techniques. We conduct an experiment that tests human position estimation without assistance versus with LV, and find that LV increases accuracy of estimating the future landing position of an object in motion about 3 times. We hope that this work will serve as a cornerstone for vision augmentation research with real-time physics prediction, and will inspire others to create new predictive visualizations.

7. ACKNOWLEDGMENTS

This work received funding from the European Union's 7th Framework Programmes for Research and Technological Development under PITN-GA-2012- 316919, EDUSAFE.

8. REFERENCES

- [1] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through HMD. In *Proceedings of ACM SIGGRAPH 1994*, pages 197–204, 1994.
- [2] A. T. Bahill and T. LaRitz. Why can't batters keep their eyes on the ball. *American Scientist*, 72(3):249–253, 1984.
- [3] R. De Charette, R. Tamburo, P. C. Barnum, A. Rowe, T. Kanade, and S. G. Narasimhan. Fast reactive control for illumination through rain and snow. In *Proceedings of IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2012.
- [4] G. Diaz, J. Cooper, and M. Hayhoe. Memory and prediction in natural gaze control. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 368(1628), 2013.
- [5] S. R. Ellis, K. Mania, B. D. Adelstein, and M. I. Hill. Generalizeability of latency detection in a variety of virtual environments. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 48, pages 2632–2636. SAGE Publications, 2004.
- [6] K. Fan, J. Huber, S. Nanayakkara, and M. Inami. Spidervision: Extending the human field of view for augmented awareness. In *Proceedings of the 5th Augmented Human International Conference*, pages 1–49, New York, NY, USA, 2014. ACM.
- [7] P. J. Hayes et al. *The naive physics manifesto*. Institut pour les études sémantiques et cognitives/Université de Genève, 1978.
- [8] Y. Itoh and G. Klinker. Interaction-free calibration for optical see-through head-mounted displays based on 3D eye localization. In *Proceedings of the IEEE Symposium on 3D User Interfaces*, pages 75–82, 2014.
- [9] Y. Itoh and G. Klinker. Vision enhancement: defocus correction via optical see-through head-mounted displays. In *Proceedings of the 6th Augmented Human International Conference*, pages 1–8. ACM, 2015.
- [10] R. Jota, A. Ng, P. Dietz, and D. Wigdor. How fast is fast enough?: a study of the effects of latency in direct-touch pointing tasks. In *Proceedings of the ACM CHI 2013*, pages 2291–2300. ACM, 2013.
- [11] H. Koike and H. Yamaguchi. Lumospheres: real-time tracking of flying objects and image projection for a volumetric display. In *Proceedings of the 6th Augmented Human International Conference*, pages 93–96. ACM, 2015.
- [12] P.-S. Laplace and A. I. Dale. *Pierre-Simon Laplace Philosophical Essay on Probabilities*, volume 13. Springer Science & Business Media, 1995.
- [13] S.-w. Leigh and P. Maes. Aftermath: Visualizing consequences of actions through augmented reality. In *Proceedings of the 33rd ACM CHI EA*, pages 941–946.
- [14] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill. Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity. In *Proceedings of the 1st Symposium on Applied perception in Graphics and Visualization*, pages 39–47. ACM, 2004.
- [15] W. R. Mark, L. McMillan, and G. Bishop. Post-rendering 3d warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 7–ff. ACM, 1997.
- [16] K. Okumura, H. Oku, and M. Ishikawa. Lumipen: Projection-based mixed reality for dynamic objects. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 699–704. IEEE, 2012.
- [17] S. Ong, M. Yuan, and A. Nee. Augmented reality applications in manufacturing: a survey. *International journal of production research*, 46(10):2707–2742, 2008.
- [18] J. Orlosky, T. Toyama, K. Kiyokawa, and D. Sonntag. Modular: Eye-controlled vision augmentations for head mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):259–268, 2015.
- [19] B. Schwerdtfeger and G. Klinker. Supporting order picking with augmented reality. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 91–94. IEEE Computer Society, 2008.
- [20] F. A. Smit, R. van Liere, and B. Fröhlich. An image-warping VR-architecture: design, implementation and applications. In *Proceedings of the ACM symposium on Virtual Reality Software and Technology (VRST)*, pages 115–122. ACM, 2008.
- [21] R. Tamburo, E. Nurvitadhi, A. Chugh, M. Chen, A. Rowe, T. Kanade, and S. G. Narasimhan. Programmable automotive headlights. In *Proceedings of 13th ECCV*, pages 750–765. Springer, 2014.
- [22] M. Tuceryan and N. Navab. Single point active alignment method (SPAAM) for optical see-through hmd calibration for ar. In *Proceedings of IEEE and ACM International Symposium on Augmented Reality (ISAR)*, pages 149–158. IEEE, 2000.
- [23] N. Umetani, Y. Koyama, R. Schmidt, and T. Igarashi. Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Transactions on Graphics*, 33(4):1–10, July 2014.
- [24] Z. Zhang, D. Xu, and M. Tan. Visual measurement and prediction of ball trajectory for table tennis robot. *IEEE Transactions on Instrumentation and Measurement*, 59(12):3195–3205, 2010.