Interference Avoidance in Multi-User Hand-Held Augmented Reality

Ohan Oda Steven Feiner

Columbia University

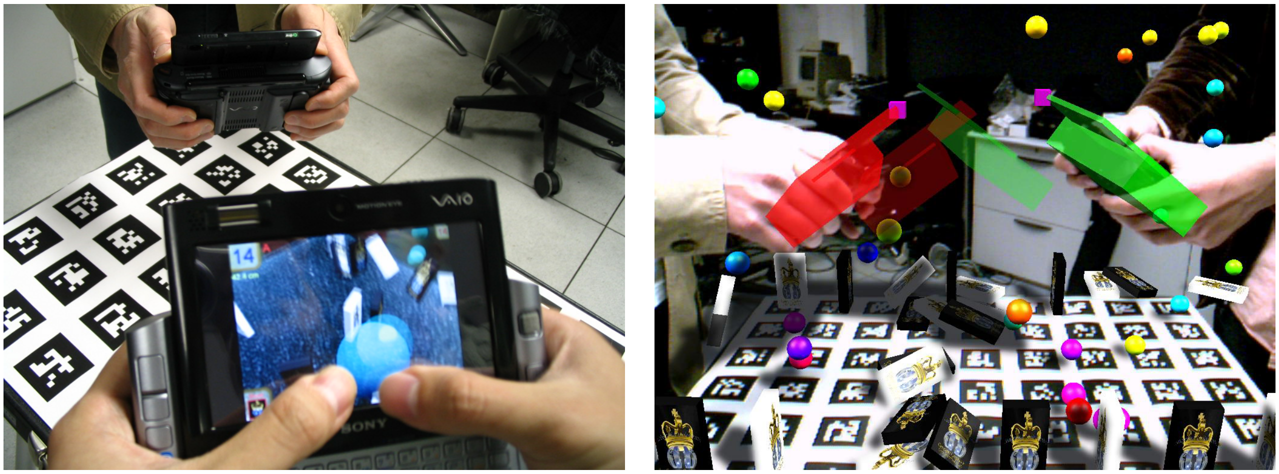

Figure 1. (Left) Player's view of two-person AR Domino Knockdown game. Virtual balls are fired at virtual dominoes by tapping on the screen. (Right) Third-person view of game with AR visualization of *Redirected Motion*. Each hand-held UMPC is overlaid with a simplified geometric model to represent its actual physical location, and a more transparent offset geometric model to represent its shifted virtual location. Two small magenta cubes highlight the two points (on the models at the physical locations) that are currently closest to each other.

ABSTRACT

In a multi-user augmented reality application for a shared physical environment, it is possible for users to interfere with each other. For example, in a multi-player game in which each player holds a display whose tracked position and orientation affect the outcome, one player may physically block another player's view or physically contact another player. We explore software techniques intended to avoid such interference. These techniques modify what a user sees or hears, and what interaction capabilities they have, when their display gets too close to another user's display.

We present *Redirected Motion*, an effective, yet nondistracting, interference avoidance technique for hand-held AR, which transforms the 3D space in which the user moves their display, to direct the display away from other displays. We conducted a within-subject, formal user study to evaluate the effectiveness and distraction level of Redirected Motion compared to other interference avoidance techniques. The study is based on an instrumented, two-player, first-person-shooter, augmented reality game, in which each player holds a 6DOF-tracked ultra-mobile computer. Comparison conditions include an unmanipulated control condition and three other software techniques for avoiding interference: dimming the display, playing disturbing sounds, and disabling interaction capabilities. Subjective evaluation indicates that Redirected Motion was unnoticeable, and quantitative analysis shows that the mean distance between users during Redirected Motion was significantly larger than for the comparison conditions.

KEYWORDS: Collaborative/competitive augmented reality, computer games, interference avoidance, computer-supported cooperative play/work (CSCP/CSCW).

INDEX TERMS: H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Artificial, Augmented, and Virtual Realities*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction Techniques*; K.4.3 [**Computers and Society**]: Organizational Impacts—*Computer-Supported Collaborative Work*; K.8.0 [**Personal Computing**]: General—*Games*

e-mail: {ohan, feiner}@cs.columbia.edu

1 INTRODUCTION

Interest is increasing in utilizing augmented reality (AR) in both competitive and collaborative social environments, such as multiplayer games and collaborative design. In these fields, an appropriate physical environment and system design are key to making the immersive experience effective and enjoyable. Except for applications in which all users are remote [2], users interact in the same physical space and with the same virtual content. Therefore, it is often possible for users to interfere physically with each other, either on purpose, when the application requires it [10], or by accident, due to lack of attention. For example, players in a multiplayer AR action game might be so engrossed that they do not notice how close they are until they actually collide. Physical interference of this sort can increase when the user's attention is focused on virtual objects rather than on other users, or when the shared environment is sufficiently small.

While some experiences encourage physical contact (e.g., the Twister board game [41] or AR games such as Human PacMan [10]), there are many social AR applications in which it would be beneficial for the application to prevent unwanted physical contact. For example, the importance of safety in game play is evidenced by the protective jackets and straps provided by Nintendo for Wii remotes [30].

In contrast to these more extreme examples, we are interested in how the system itself can discourage unwanted and unintentional

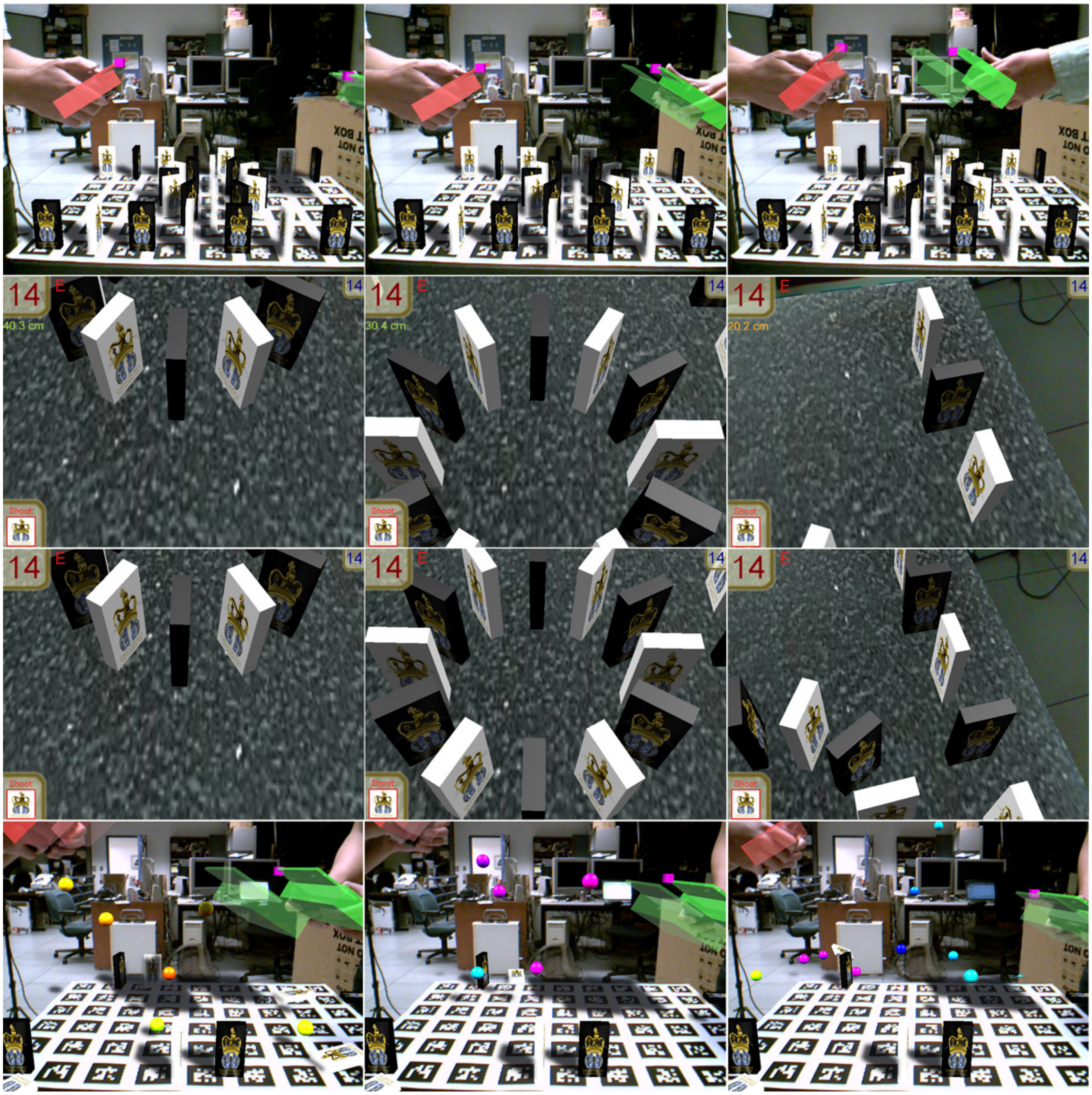

Figure 2. (Row 1) Changes in physical and virtual locations under Redirected Motion. As the green player moves toward the red player, the virtual location of the green player is shifted along the green player's direction of movement, while the (stationary) red player's virtual location is not affected. (Row 2) First-person screenshot views for the green player *with* Redirected Motion, corresponding to images on row 1. (Row 3) First-person views for the green player *without* Redirected Motion (i.e., no virtual shifting), corresponding to images on row 2. (Row 4) As the green player moves back beyond the distance threshold $\boldsymbol{\tau}$ from the red player, the virtual location of the green player is shifted back to its physical location. Representations of physical and virtual locations and closest points are the same as in Figure 1.

physical interference without making users aware that this is happening. We address *hand-held AR*, in which each user has a hand-held device that displays information based on its tracked position and orientation, as shown in Figure 1. We are primarily interested in avoiding collisions between these tracked devices (and, as a side effect, in avoiding collisions between users, who might otherwise not be tracked).

We refer to the location (position and orientation) of a user's device in the physical world as the user's *location.* We will sometimes refer to the user's location as the user's *physical location* for emphasis, and will often make an implied reference to the user's location just by referring to the *user*. In contrast, we will use the term *virtual location* to refer to a possibly different location at which the system treats the device as being for purposes of rendering and interaction. We make three assumptions:

1. Users are most interested in the location toward which they are moving and the virtual objects with which they can interact at that location.
2. Users are aware of each other's physical location.
3. Users are not intentionally trying to physically interfere with each other.

We have developed an interference-avoidance technique, Redirected Motion that attempts to address these assumptions. As shown in Figure 2, *Redirected Motion* transforms the virtual location of a user as they approach another user to keep the users from colliding, inspired by earlier work on redirected walking in virtual

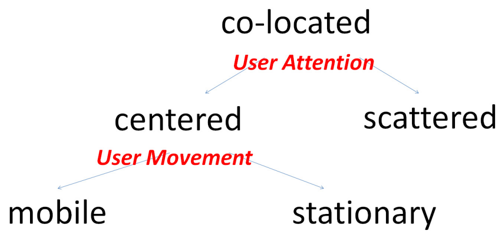

Figure 3. Classification of co-located environments.

reality (VR) [37]. When user *A* reaches a set threshold distance from user *B*, then, as *A* continues to move toward *B*, *A*'s virtual location is shifted ahead of *A*'s physical location. The shift occurs in *A*'s instantaneous direction of travel, where the amount of shift is proportional to *A*'s velocity. The intent is that by exaggerating *A*'s travel virtually, *A* could be prevented from physically contacting *B* when *A*'s destination is sufficiently close. In other words, we hypothesize that shifting *A*'s virtual location ahead of *A*'s physical location could, in many cases, cause *A* to "stop short," and, thus, could keep the physical distance between *A* and *B* from being as small as it would be if this technique were not applied.

Note that since Redirected Motion offsets the location from which virtual objects are rendered, it is not suitable for applications in which the offset between the virtual and physical locations will violate registration requirements. The challenge is how to avoid perceived misregistration. We address this in two ways. First, the application shown in Figures 1–2 is typical of many AR games in which there is a relatively loose connection between the virtual and physical worlds, with the physical world made visible largely to support social interaction. Like some examples of this genre (e.g., [20, 29]), our game is also played using a tracked board that is completely covered with a virtual texture, while allowing the rest of the environment to remain visible. In our case, this virtual texture prevents players from seeing virtual game objects appear to translate relative to the real board while the offset between the rendered virtual view and the real camera view changes. Second, as we discuss later, we also place an application-dependent limit on the magnitude of the offset.

In the remainder of this paper, we first discuss related work. Then, we explain Redirected Motion in more detail and describe several alternative interference-avoidance techniques. Next, we present the design and results of a formal user study that we conducted to measure how effective each of the techniques is at keeping users apart and how distracting each technique is. Finally, we discuss limitations and applications of Redirected Motion, state our conclusions, and describe future work.

2 Related Work

Existing social AR systems can be roughly classified into three categories in terms of interaction space: indoor co-located (e.g., [3, 5, 6, 12, 13, 18, 20, 23, 27, 32-35, 38, 40, 44]), indoor remote (e.g., [2, 14, 22, 24]), and outdoor (e.g., [4, 7, 10, 11]). We can further classify co-located environments into subcategories, as shown in Figure 3. When users' interaction spaces are remote, there can be no physical interference between the users. However, physical interference can occur when users are co-located and their attention is either scattered around the environment (e.g., [6, 12, 27]) or centered in one location and they are expected to move around in the entire application space (e.g., [3, 32, 44]). For example, the non-driver players in our AR racing game [32] can move arbitrarily around the game board to place tracked obstacles in desired places. In contrast, when users are expected to stay in one location (e.g., [13, 18-20, 34, 40]), it is unlikely for them to interfere with

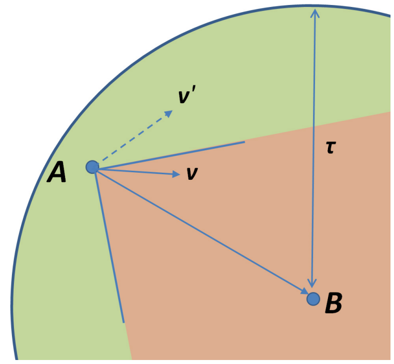

Figure 4. User *A*, moving with velocity *v*, is treated as moving toward user *B* only if the angle between *v* and the vector from *A* to *B* is less than 45° (the red area). For example, the solid vector *v* is classified as moving toward user *B* and the dashed vector *v′* is not.

each other physically. For example, each of the two players in AR tennis [19] may move around on the side of their court, but is not allowed to cross the net, eliminating the possibility of physical interference. In this paper, we will focus on the case where users are co-located and are expected to move around in the entire application space.

Researchers have long explored how to prevent users from colliding with physical obstacles, such as walls, in virtual (and real) environments. One early method plays a warning sound when the user comes within a preset distance from an obstacle. Cheok and colleagues [12] apply a more subtle version of this approach in their social AR application, Touch-Space: The spatialized 3D sound of an airplane propeller is used to make each player aware of the other player's location, even when they cannot be seen.

Fitzpatrick and colleagues [16] take advantage of the decisive role the balance mechanism of our inner ears play in directing the human walk; they show how applying an electrical current to a surface electrode placed behind the ear can be used to remotely "steer" a blindfolded user as they walk, without the user noticing the redirection if it is sufficiently subtle. In contrast, Razzaque and colleagues [37] have developed "redirected walking," in which the virtual location of a user in VR is transformed by injecting additional virtual rotation, especially while the user is physically moving. This can change the direction in which the user walks, cumulatively affecting the user's physical location and steering the user away from obstacles—without the user realizing it if the incremental modifications are sufficiently small. While redirected walking assumes that the user is in VR, Redirected Motion was inspired by this idea of transforming a user's virtual location to influence their physical location.

Other work has explored scaling interaction space in VR, For example, virtual motion can be scaled up from physical motion to allow the user to traverse virtual space faster [8, 21] or extend their virtual reach [36]. However, the use of these techniques is typically intended to be perceptible to the user. There is also an analogy to "mouse acceleration" in 2D desktop user interfaces, in which the mouse transfer function is dynamically modified to increase the speed of the cursor nonlinearly as the mouse is moved faster [26]. Here, the tradeoff between precision and speed is of concern, not avoiding interference.

Finally, we note that collision avoidance is well-studied in multi-robot interaction [1, 9, 17]. However, the approaches adopted for that domain are quite different, typically involving direct control over the robot's motion without any concern for keeping the robot "unaware" of the change.

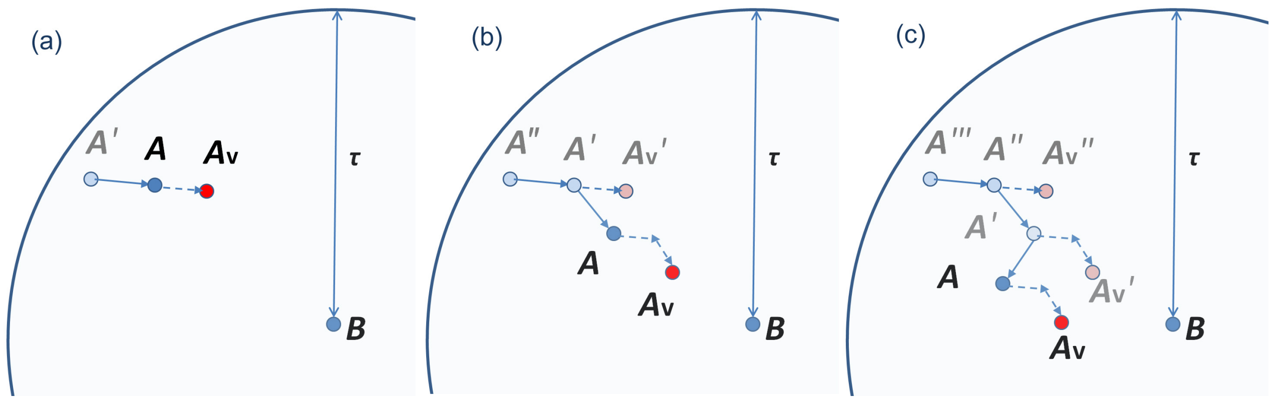

Figure 5. Physical and virtual locations of user *A* over three frames. In each frame, *A* indicates the current physical location, and *A*v indicates its shifted virtual location. Primed labels indicate previous physical and virtual locations. Solid vectors indicate motion of the physical location of A. Dashed vectors are components of the virtual shifting vector $\vec{G}$, exaggerated for this figure. (a–b) Additional virtual shift is applied along the motion vector from *A'* to *A*, since the user moves toward *B*. (c) Additional virtual shift is not applied, since the user does not move toward *B*.

3 REDIRECTED MOTION

Using Interrante and colleagues' [21] idea of scaling distance traveled by amplifying the user's velocity only in the direction in which they are walking, we apply an additional translation to the virtual location of a user in the direction in which they are moving. Note that this transformation will affect just the overlaid virtual scene. As shown in Figure 4, the translation is applied to user *A* only when their distance to another user *B* is under a specified threshold **τ** and *A* is moving toward *B*. To capture the notion of moving toward B as more than just getting closer to B, we include only motion in which the velocity component in the direction from *A* to *B* is greater than the component perpendicular to that direction.

$\vec{G}$ is the 3D *virtual shifting vector*, the translation from a user's physical location to their virtual location. $\vec{G}$ is initially (0,0,0), and is incremented at each frame by adding $\vec{g}$, which is calculated as:

$$\vec{g} = \frac{\delta \cdot \vec{v}}{M_1}, \tag{1}$$

where $\vec{v}$ is the user's motion vector, and M_1 is a scalar constant to adjust the shifting distance. δ is 1 when the user moves more than a certain motion threshold ψ ($|\vec{v}| > \psi$) and the distance between the users is less than **τ**, otherwise, δ is 0. When the user is outside the distance threshold **τ**, $\vec{G}$ is gradually shifted back to (0,0,0) by μ cm each frame that the user moves more than ψ.

The elements *x*, *y*, and *z* of $\vec{G}$ are clamped between lower and upper shifting limits along each axis, such that $x_{lower} \le x \le x_{upper}$, $y_{lower} \le y \le y_{upper}$, and $z_{lower} \le z \le z_{upper}$. We do this to limit the amount of visual mismatch between the virtual view and physical view. In Section 6, we provide the specific values used for these parameters in our study application, as decided based on pilot studies. Figure 2 shows the technique as applied in our study, while Figure 5 shows schematic examples of how the physical and virtual locations of a user change over three frames.

Note that there is a clear distinction between this approach and a simpler one in which the virtual location is offset from the front of the device by a fixed vector throughout the application (a 3D analogue to Vogel and Baudisch's [42] shifting of the 2D selection point on a touch-screen a set distance away from the user's finger). If the virtual location is always offset by the same amount, then there will be always a mismatch between the virtual and physical worlds, even when one is not needed. Furthermore, the offset will not reflect the direction in which the user is moving or the direction to the other user. For example, offsetting from the front of the device may not be useful when the users are standing shoulder to shoulder, with their devices facing in the same direction.

4 OTHER INTERFERENCE AVOIDANCE TECHNIQUES

We were interested in comparing Redirected Motion with other techniques that are intended to persuade players to move away from each other when in physical proximity. The techniques we describe in this section modify what a user sees or hears, and what actions they can perform, but do not create an offset between the user's physical and virtual location. Instead, the goal is to alert the user to the impending interference, under the assumption that this will cause them to modify their trajectory.

The techniques are applied only when the distance between the users' physical locations is under a preset threshold **τ′**. Here, we choose **τ′** to be lower than **τ**, to minimize distraction, since these techniques are intended to be noticeable. As with Redirected Motion, applying the techniques to a user is based on the absolute velocity of that user. (Thus, a user who is not moving toward another user will never have their virtual location offset from their physical real location, no matter how close the other user is or how fast the other user approaches.) We discuss the specific parameters used in our implementation in Section 6.

4.1.1 Screen Dimming

Our first alternative technique modifies what the user sees by dimming their display. This alerts the user, but can also be punitive, since the dimmer the screen, the less well the user can see the virtual objects and perform effectively. The amount by which the screen is dimmed at each frame is computed as:

$$f = \frac{\delta \cdot |v|}{d \cdot M_2} \qquad (0 \le F \le 1)\,. \tag{2}$$

The current brightness of the screen, *F,* to which *f* is added at each frame, is clamped between 0 and 1, where 0 means normal brightness and 1 means totally off. $|\vec{v}|$ is the movement speed, *d* is the distance between the users, M_2 is a constant that determines how rapidly the brightness changes, and δ is either 1 in the case the user is moving toward the other user, or −1 if not. The screen brightness changes only if the user moves more than a motion threshold ψ′ ($|\vec{v}| > \psi'$) when the distance between the users is less than **τ′**. The screen gets dimmer and dimmer as the user moves their display toward the other player, until it is totally off. The screen gets dimmer quicker as *d*, the distance between the users, decreases. Since the technique is not intended to be imperceptible, *F* is reset to 0 once the user moves back beyond the **τ** distance threshold.

4.1.2 Sound Beeping

There are many ways in which audio can be used to notify the user of the possibility of interference. A sound can be played if the user is within **τ′**, and various properties of the sound can be predicated on the user's velocity (e.g., modifying frequency, amplitude, or waveshape). In contrast to the more subtle use of spatializd 3D audio in Touch-Space [12], we use a simple "beep." While the user is within the distance threshold **τ′**, the number of beeps per second is incremented by one every ρ msecs if the user is moving toward another user and decremented by one every ρ msecs otherwise, if the movement speed $|\vec{v}|$ is greater than the movement threshold ψ′. The number of beeps per second is clamped between 0 and ω. The frequency is reset to 0 once the user moves back beyond the distance threshold **τ′**.

4.1.3 Action Disable

Disabling user interaction capabilities is also an effective way of modifying user experience. Inspired by the punitive response of a pinball machine that has been "tilted" by the user, the system prevents user interaction. This is a Boolean operation in our implementation of this approach: we disable user action (in this case, preventing balls from being fired) as soon as the user crosses the distance threshold **τ′**. Action is immediately enabled after the user moves out of the distance threshold.

5 Test Application

We developed a two-player game, AR Domino Knockdown, shown in Figures 1–2, in which to test our techniques. Each player attempts to knock all of the other player's virtual dominos off a real table (45 cm wide × 65 cm long) by shooting virtual balls with a hand-held ultra-mobile personal computer (UMPC) through which they view the environment. The table is covered with an array of fiducial markers for tracking, and we overlay a virtual texture resembling the table's surface on top of the marker array. Two sets of virtual dominos, each with a distinctive color and texture, are initially placed in a startup configuration on the table.

Each player shoots virtual balls by tapping their fingers on the screen of their UMPC, which launches virtual balls from the tapped locations. Players can move anywhere around the table and shoot from any angle they desire, as long as at least one marker in the array on the table is visible from and can be tracked by the device's embedded camera. The device's position, orientation, and instantaneous velocity determine the direction in which a ball is fired and its speed. A rigid-body physics simulation is performed on the virtual balls and dominos. When a domino falls off the table, it is removed from the simulation. The numbers of dominos that each player has left is displayed on the screen, along with the distance between the players. The color of the distance display is initially presented in yellow green, and if the distance gets below the threshold **τ′**, it turns orange.

5.1 Software

Our game is built using Goblin XNA [31], a framework for research on VR and AR, with an emphasis on games. It is based on Microsoft XNA Game Studio 3.0, and written in C#. The framework supports 6DOF (six degrees of freedom) position and orientation tracking using optical marker-based tracking through ALVAR [43] and ARTag [15], in addition to providing a 3D scene graph, rigid body physics simulation (using Newton Game Dynamics [28]), networking (using Lidgren [25]), shaders, particle systems, and 2D user interface primitives.

Players communicate through a central server for which the player machines are clients. We use a server-based model instead of a peer-to-peer model to equalize and minimize the load on the clients. The physical simulation is performed on the server, and the simulated results are broadcast to the clients so that both see the exact same result. The central server broadcasts the game status in addition to the simulated results, and the client transmits its camera transformation (which is the inverse transformation of the marker array it sees on the table) and the shifting transformation if Redirected Motion is used to the server.

Table 1. Summary of parameters used in user study.

Symbol	Value	Description
τ	41 cm	Distance threshold between users *A* and *B*, below which Redirected Motion is triggered
τ′	25 cm	Distance threshold between users *A* and *B*, below which interference avoidance techniques other than Redirected Motion are triggered
$\{x_{lower}, y_{lower}, z_{lower}\}$	{−4.5 cm, −6.5 cm, 0 cm}	Lower bounds of shifting vector elements for Redirected Motion
$\{x_{upper}, y_{upper}, z_{upper}\}$	{4.5 cm, 6.5 cm, 0 cm}	Upper bounds of shifting vector elements for Redirected Motion
μ	0.01 cm	Shifting amount applied every ρ msecs when shifting $\vec{G}$ back to (0, 0, 0)
ψ	0.3 cm	Lower motion velocity threshold for enabling Redirected Motion when the physical distance between users *A* and *B* is under **τ**
ψ′	0.3 cm	Lower motion velocity threshold for triggering interference avoidance techniques other than Redirected Motion when the physical distance between users *A* and *B* is under **τ′**
M_1	2	Scalar constant to adjust shifting distance
M_2	3	Scalar constant that determines how rapidly brightness changes for Screen Dimming
ω	30	Average frames per second for application
ρ	33	Msecs between render frames

5.2 Hardware

The application runs on Sony VAIO UX-VGN-380N and UX-VGN-390N UMPCs, which are 1.2 lb., 5.9" (W) × 3.7" (H) × 1.3" (D) hand-held devices with a 4.5" diagonal touch-sensitive LCD screen and an integral camera in the back of the device. We use two of these for the user study. Both of them run Windows XP on a 1.33 GHz Core Solo CPU with 1 GB memory, and an Intel

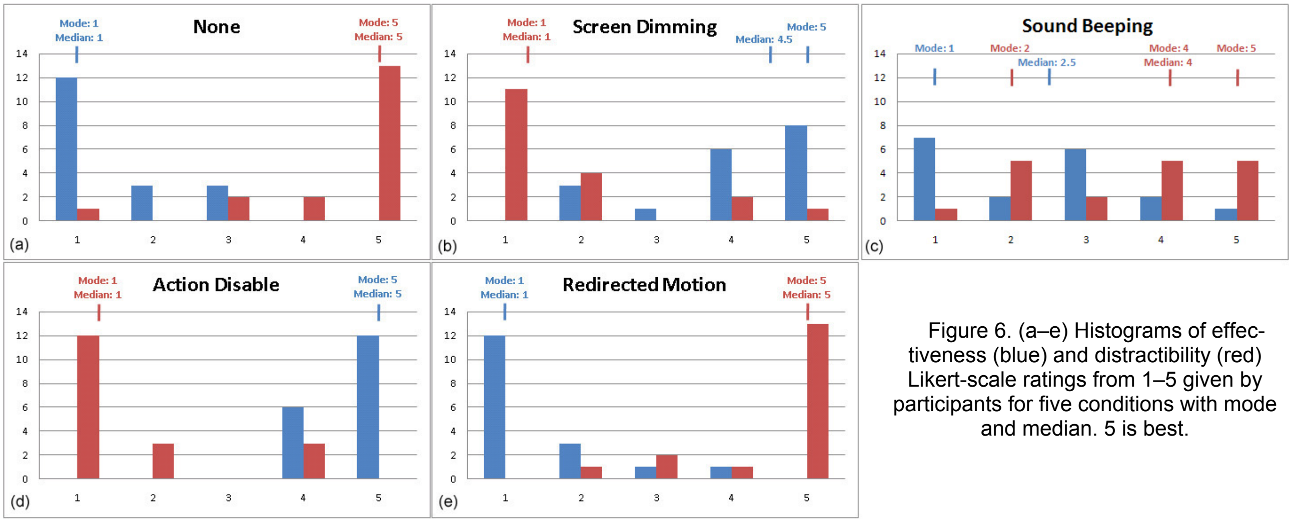

Figure 6. (a–e) Histograms of effectiveness (blue) and distractibility (red) Likert-scale ratings from 1–5 given by participants for five conditions with mode and median. 5 is best.

945GMS graphics chip. Both UMPCs communicate through a dedicated WiFi network with a laptop that acts as a central server. The laptop is a Sony VAIO VGN-SZ480 running Windows Vista with 2.33 GHz Intel Core 2 Duo CPU, 2 GB memory, and NVidia GeForce Go 7400.

6 PILOT STUDIES

Prior to conducting the formal user study, we performed informal pilot studies with other lab and department members in order to choose suitable values for the variables used in the equations and thresholds specific for our experimental setup, as shown in Table 1. We determined that $M_1 = 2$ would produce unrecognizable, yet significant shifting for Redirected Motion, $M_2 = 3$ would render visually smooth transition for dimming, and $\boldsymbol{\tau'} = 25$ cm (~10 inches) would be a reasonable distance for the effect threshold. As the player taps on the screen to shoot balls, it moves the display slightly, which we noticed can be up to **~** 0.3 cm. Therefore, we set both ψ and ψ′ to be 0.3 cm. Shifting by 0.01 cm every ρ msecs when the user's movement speed was more than 0.3 cm/sec was fast enough, as well as unnoticeable, so we set μ = 0.01 cm. Since our application was running around 30 frames per second, we set ρ = 33, and ω = 30.

To determine the shifting distance threshold $\boldsymbol{\tau}$, we first needed to determine lower and upper shifting bounds for our game. Since we did not want the participants to realize that the view point was shifted, we decided to overlay the table with a texture size larger than the actual table. If the texture size is exactly same as the table size, then it would become obvious that the viewpoint was shifting. Either the dominos would need to shift relative to the texture (which would look odd, especially toward the edge of the table) or the texture would have to shift too, uncovering the table as it was translated, and clearly moving, as compared to the edges, since we cannot shift the physical view seen by the camera. To compensate for this anticipated mismatch between the virtual and physical view, we decided to overlay the tabletop with a texture that is larger than the actual tabletop, but not so big that the participant could easily realize the discrepancy. By setting the shifting limit to (texture size – actual size) / 2, the physical table will never be uncovered. Based on our pilot studies, we decided to use a texture 1.2 times larger than the actual size of the table (1.2 × {45cm × 65cm} = {54cm × 78cm}). Thus, $\{x_{lower}, y_{lower}, z_{lower}\} = \{-4.5\text{cm}, -6.5\text{cm}, 0\}$, and $\{x_{upper}, y_{upper}, z_{upper}\} = \{4.5\text{cm}, 6.5\text{cm}, 0\}$. We do not shift the virtual location along the *z*-axis, since we want to have the stationary dominoes to appear to be resting exactly on top of the physical table. (If the majority of virtual objects were not supposed to reside on the physical table, we would have shifted in *z*, as well.)

Once we decide on the shifting limits, we could compute the maximum shifting amount as $\sqrt{4.5^2 + 6.5^2} \approx 8$ cm. Since $M_1 = 2$, a motion distance of 8 cm × 2 = 16 cm is required for full shifting. Since we wanted to maximize the effectiveness of Redirected Motion, we wanted the participants to be shifted for at least half the maximum amount in the worst case before they are considered to be too close to each other. In the worst case, when players move toward each other, the total distance they move will be approximately 16 × 2 = 32 cm before they get fully shifted. Thus, since we had chosen $\boldsymbol{\tau'} = 25$ cm, we decided to set $\boldsymbol{\tau} = 25$ cm + 32 / 2 cm = 41 cm.

7 USER STUDY

To understand how well our avoidance techniques work in terms of effectiveness and distraction level, we conducted a formal user study. 18 paid participants were recruited by flyers and mass email requesting that users apply as pairs (dyads). All participants were university students (3 female and 15 male, 22–36 years old, average 26 years old). All subjects use computers multiple times per day, but only two had previous experience with AR, and only one subject had previously used a UMPC.

The members of each dyad were asked to play multiple rounds of the two-player AR Domino Knockdown game under different test conditions, as controlled by a test scaffold. We measured the effectiveness and distraction level both qualitatively and quantitatively. Qualitative measurement was assessed through a questionnaire. Quantitative measurement of effectiveness was assessed by computing the average distance between the two players during game play. The distance between the two players was recorded at each frame as the shortest distance between any point on one UMPC and any point on the other UMPC, computed by the physics engine using the 3D models of the UMPCs at their physical locations, as shown in Figure 1. We also recorded the average completion time of each game round as a potential measurement of distraction level.

7.1 Study Description

Our user study was a within-subject, single-session experiment in which we compared each participant's behavior on a single task using four different interference avoidance techniques plus a control condition in which none of the techniques were used. We began each dyad's session by showing the two participants how to play the game.

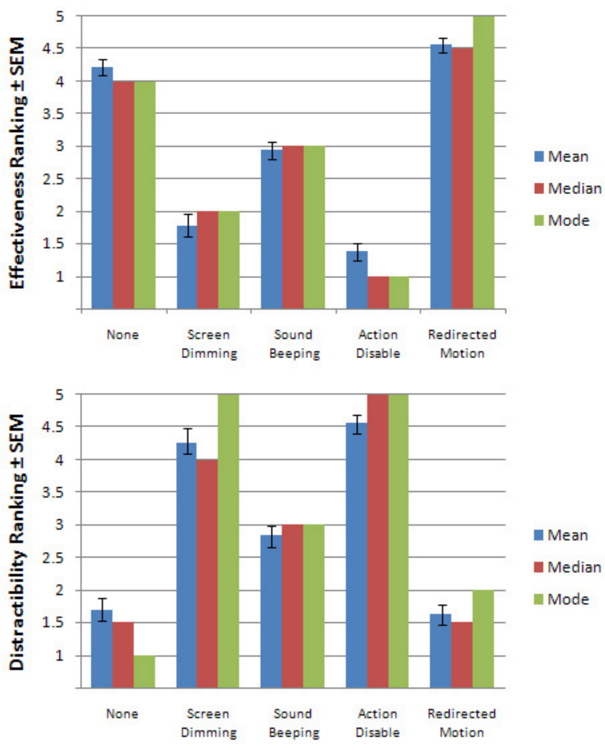

Figure 7. Ranked qualitative effectiveness (top) and distractibility (bottom) with SEM (Standard Error of the Mean) for the five technique conditions (including None). 1 is best (perceived to be most effective, least distracting).

Participants played a total of 25 rounds, with each round typically taking 30–120 seconds. Each round was initiated by the study coordinator so that the participants could rest or provide comments as desired between rounds. Participants initially played one practice block of five rounds with none of the techniques enabled to get acclimated to the game, followed by four blocks of actual user study. Each of the four study blocks contained five rounds (one each for the four different interference avoidance technique and one in which no interference prevention techniques were used). Technique order within the study blocks was randomized using a Latin square.

Before participants started the first (practice) block, the study coordinator told each participant the color (white or black) of their designated target dominos, which remained the same throughout the study. The two participants were asked to position themselves at opposite ends of the table along the longer edge (65 cm) at the start of each round, and were reminded to avoid shooting their own dominos as much as possible and to move around the table to shoot more precisely. Before participants started the remaining blocks, the study coordinator informed them that they would see certain effects if they got too close to each other, and that they would need to move away from each other if they wanted the effects to disappear. They were also told that some effects are quite obvious, but some are very subtle. To avoid biasing participants between control rounds and Redirected Motion rounds, they were not told that each block contained a control round in which no technique was used.

We found that the players during our pilot studies were not exactly sure when the new game round starts after the study coordinator signalled the server to move on to the next round, so we provided a countdown signal with sound and text message at the beginning of each round, and asked the players to start shooting after the countdown ended.

Table 2. Paired Wilcoxon test results on perceived effectiveness (top) and distractibility, where N=None, SD=Screen Dimming, SB= Sound Beeping, AD=Action Disable, and RM=Redirected Motion.

	N-SD	N-SB	N-AD	N-RM	SD-SB	SD-AD	SD-RM	SB-AD	SB-RM	AD-RM
					Effectiveness					
Z	3.55	3.05	3.55	1.60	2.97	1.44	3.53	3.59	3.53	3.57
p	.000	.002	.000	.109	.003	.151	.000	.000	.000	.000
					Distractibility					
Z	3.42	2.80	3.44	0.27	2.89	0.91	3.37	3.45	3.25	3.43
p	.001	.005	.001	.791	.004	.361	.001	.001	.001	.001

Each participant was asked to fill out a post-hoc questionnaire. The questionnaire was designed to assess the effectiveness and distraction level of the five conditions using Likert-scale questions ranging between 1 and 5 (1 = worst, 5 = best), a request to rank the five conditions with ties allowed, and space for free-form comments. To avoid the bias that could result from using descriptive technique names, we labeled each condition alphabetically (A–E). These letters were displayed on the UMPC screen for each game round during the actual user study, so that the participants could associate each effect with each letter. The study coordinator also emphasized which letter maps to which technique and explained the meaning of effectiveness right before each participant was asked to fill out the questionnaire. Participants were told that the effectiveness of a technique referred to how well that technique kept the two participants away from each other. Since we expected that participants would not detect any effect for Redirected Motion because of attempts to make it unrecognizable, the study coordinator also repeatedly asked the participants what effects they saw for the control condition and Redirected Motion. Participants were also asked to remember the difference between the control condition (A) and Redirected Motion (E) to avoid confusion when completing the questionnaire.

Total time for the study was approximately one hour. During the study, the software collected game play data including geometric relationships between the players' UMPCs (computed from the tracked 3D positions and orientations of the UMPCs), game scores and scoring profiles, duration of the time that players were within threshold $\boldsymbol{\tau}'$, and game duration.

7.2 Hypotheses

Prior to our experiment, we formulated five hypotheses:

H1: The control condition (None) will be least effective and least distracting.

H2: Sound Beeping will be less effective, but less distracting than Screen Dimming or Action Disable, since the player can simply ignore the beep.

H3: Screen Dimming and Action Disable will be most effective, but most distracting, compared to other techniques, since visibility or interaction capability are impaired.

H4: Redirected Motion will be as effective as Screen Dimming or Action Disable and much more effective than Sound Beeping or None.

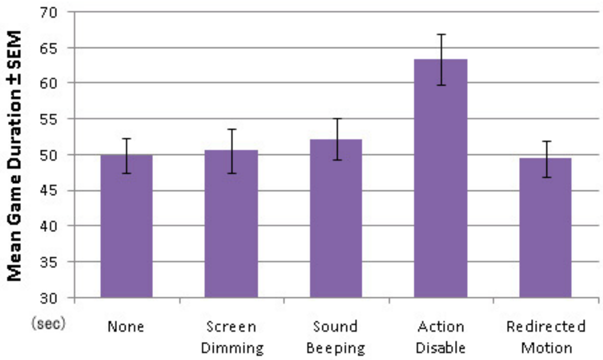

Figure 8. Mean game duration with SEM.

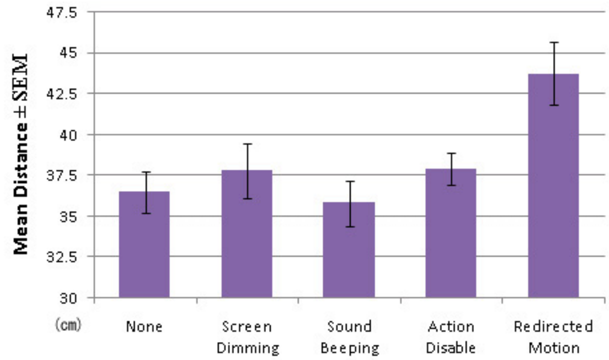

Figure 9. Mean distance between participants' devices with SEM.

H5: Redirected Motion will be the least distracting compared to Screen Dimming, Action Disable, or Sound Beeping, since the players do not perceive the shifting effect.

Overall, we expected to show that Redirected Motion would be the most suitable interference avoidance technique, since it would be quite effective, yet not distracting.

7.3 Analysis

We ended up not using the data from the first study block (i.e., the second of the five blocks) because it was clear that several participants were perplexed by the perceptible techniques when they first encountered them, mitigating learning curve bias for these first encounters. Therefore, we analyzed 27 game rounds (9 dyads × 3 study blocks) for each of the five technique conditions. We analyzed our results according to Likert-scale ratings, subjective ranking, game duration, and mean distance between the participants, using $\alpha = 0.05$ as our criterion for statistical significance.

7.3.1 Qualitative Analysis

A Friedman test shows that the distribution of the rankings (Figure 7) for perceived effectiveness ($X^2_{(4)}$=51.54, p<0.001) and distractibility ($X^2_{(4)}$= 45.87, p<0.001) between the different techniques is significant. A Wilcoxon test (Table 2) indicates that None and Redirected Motion have significantly less perceived effectiveness and less distractibility compared to other three techniques. Multiple participants mentioned "I didn't notice anything" for both techniques. Since most users did not perceive any difference between None and Redirected Motion, this does not confirm H1. Sound Beeping was evaluated to be significantly less effective, but less distracting compared to Screen Dimming or Action Disable, as shown in Table 2, qualitatively validating H2. Both Screen Dimming and Action Disable were evaluated as significantly more effective, but more distracting than the other conditions, as shown in Table 2, which qualitatively validates H3.

The Likert-scale results (Figure 6) show that participants found Screen-Dimming and Action-Disable technique to be quite effective, but very distracting. Sound Beeping was not perceived to be very effective, as participants mentioned that they could simply ignore the sound and keep playing. Participants thought that None and Redirected Motion were not distracting. This validates H5. We address hypothesis H4 in the quantitative analysis, since participants were not aware of the effect.

Even though most participants did not notice any difference between None and Redirected Motion, one participant said that he perceived a slight increase in gravity that caused the balls to land on the table closer to him compared to other techniques and another participant said that he noticed that the ball was fired slightly off from the location he tapped on the screen. These arguments are understandable, since dynamically changing the displacement of the virtual location from the physical location could cause these small discrepancies.

7.3.2 Quantitative Analysis

A within-subjects one-way ANOVA shows that technique had a significant effect ($F_{(4,23)}$=5.382, p<0.01) on game duration (Figure 8). A paired-sample *t* test between Redirected Motion and Action Disable ($t_{(26)}$=3.611, p<0.01) shows that rounds played with Redirected Motion take significantly less time to complete compared to ones played with Action Disable, but there was no significant difference between Redirected Motion and other techniques. Paired-sample *t* tests confirm that rounds played with Action Disable took significantly longer than ones played with all other conditions—None: ($t_{(26)}$=4.628, p<0.01), Screen Dimming: ($t_{(26)}$=3.817, p<0.01), Sound Beeping: ($t_{(26)}$=3.646, p<0.01). We believe that this occurs because when action is disabled, dominos could not be knocked off the board. This increase in game duration is consistent with the high level of perceived distractibility for Action Disable shown in the Likert-scale results and rankings.

To quantitatively analyze technique effectiveness, we computed the mean distance between the closest points of the participants' devices during game play. Before analyzing the distance data, we cleared them by removing outliers and by ignoring the data collected during the four second countdown period at the beginning of each game round. Distances less than 1 cm or larger than 120 cm (1.5 times the diagonal of the table) were considered to be outliers because it is most likely that any distance outside of this range was caused by an (infrequent) tracking glitch. Tracking glitches could happen when the players were either so close to each other that a UMPC or body part could block the other player's UMPC camera or too far away from the marker array on the table to track it. Outliers accounted for 0.7% of all distance data and were present for all five techniques: 0.21% of None, 0.21% of Screen Dimming, 2.7% of Sound Beeping, 0.31% of Action-Disable, and 0.6% of Redirected Motion. (There were more outliers for Sound Beeping than for other conditions because in one round played with Sound Beeping, one of the participants moved quite far from the table, resulting in a relatively large number of tracking glitches.)

We asked the participants not to move from their initial positions at the opposite ends of the table until the four-second countdown finished, but several started to move closer to the board before the countdown ended, after they got accustomed to the game. Therefore, in order to remove this initial bias across the user study, we ignored the distance data collected during the countdown.

A within-subjects one-way ANOVA on the cleared data shows that technique had a significant effect ($F_{(4,23)}$=5.504, p<0.01) on distance between the participants over time (Figure 9). Paired *t* tests between Redirected Motion and the other conditions showed that during Redirected Motion, participants maintained a significantly larger mean distance between them, especially comparing Redirected Motion and None, and Redirected Motion and Sound Beeping—None ($t_{(26)}$=3.946, $p<0.001$), Screen Dimming ($t_{(26)}$=3.414, $p<0.01$), Sound Beeping ($t_{(26)}$=4.462, $p<0.001$), Action Disable ($t_{(26)}$=2.854, $p<0.01$). This validates H4. However there was no significant difference among the comparison conditions.

That these other conditions did not significantly increase average distance between participants means that we did not quantitatively confirm the other hypotheses about effectiveness: H1–H3. Observations made during the user study indicate some possible explanations for the overall ineffectiveness of these perceptually obvious techniques. While some participants appeared to stay farther away from their opponent when the other techniques were used, other participants appeared to try to use these techniques strategically against their opponent. Although, we noted in Section 4 that the techniques only affected a participant who was actively moving (relative to the game board) toward their opponent, once both participants were within distance $\boldsymbol{\tau}'$, any motion toward the other by either participant would elicit the effect. Thus, an aggressive player could take advantage of the effect to try to limit the utility of their opponent moving in their direction. Furthermore, with the exception of Action Disable, a player could either ignore the beeping or plan an attack at a safe distance and then move in briefly to execute it with a dimmed screen.

8 Limitations And Applications

As we pointed out in Section 1, Redirected Motion is not suitable for AR systems in which offsetting virtual objects from physical objects would violate requirements for accurate registration between those virtual and physical objects. These include applications in which virtual objects must be in close proximity to visible physical objects whose appearance would make the offset obvious. Since the application used in this paper employed a patterned physical game board on which virtual objects were placed, we avoided this situation because we used the well known technique of overlaying the physical game board with a virtual texture. If the virtual objects were sufficiently distant from physical objects (e.g., if they were floating high enough above the game board), it would have been possible to avoid adding the virtual game board texture. Based on our pilot studies, we also established bounds on the amount by which a user's virtual location can be shifted relative to their physical location, and the speed with which the shift is made.

We believe that Redirected Motion would also be applicable to AR applications other than games. For example, consider a collaborative AR environment in which co-located users design a product by constructing virtual parts and assembling them in a shared space. When multiple users work on the same part or assembly, Redirected Motion could make it possible for one user to view and interact with it from a virtual location that would be uncomfortably close to (or even physically blocked by) another user. Our experiment intentionally tried to avoid making participants aware that Redirected Motion was being used, In a collaborative AR environment, however, we might want all users to know when a user's physical and virtual locations were different. This might be accomplished by offering the ability to view the other users' virtual locations as ghosted overlays similar to those used in Figure 2.

9 Conclusions

We have presented a software technique, Redirected Motion, that can prevent physical interference between users in appropriate collaborative or competitive AR environments. Redirected Motion is similar in spirit to the redirected walking technique used in VR to avoid physical collision, but addresses a multi-user hand-held AR environment by injecting translation rather than rotation. While redirected walking repositions the user away from physical boundaries, our technique relocates the users away from each other physically by moving them closer virtually.

To validate that Redirected Motion is a suitable technique for avoiding physical interference in AR without degrading user performance, we conducted a formal user study. We compared this technique against a control condition in which no technique is used, as well as several techniques used for warning users of potential interference. The study showed that Redirected Motion was significantly more effective than when no technique was used, more effective than the comparison conditions, and as nondistracting as when no technique is used.

10 Future Work

In this paper, we studied two-player hand-held AR interactions. A more general equation may need to be devised for more than two users. Even though Redirected Motion is designed for avoiding interference between users, we also believe that it could be used to avoid passive obstacles.

For our distance thresholds $\boldsymbol{\tau}$ and $\boldsymbol{\tau}'$, we used a single constant value for the entire duration of the application. However, depending on the orientation of the UMPC (and its integral camera), the distance to be considered as too close may change because the player's physical body is asymmetric relative to the UMPC and its camera. For example, if the users are facing each other, then since their bodies are typically behind their cameras, $\boldsymbol{\tau}'$ can be smaller. However, if the users are next to each other with their displays approximately coplanar, then the players' arm lengths need to be considered in addition to the geometry of the UMPCs themselves, so $\boldsymbol{\tau}'$ should be larger. Thus, it may be more appropriate to make $\boldsymbol{\tau}$ and $\boldsymbol{\tau}'$ vary depending on the camera orientation.

We did not explore rotational gain, since we speculated that it would not provide as much of a distance buffer as translational gain; however, for certain situations, especially when the players do not need to move much, rotational gain may work better than translational gain. For example, if the players are next to each and required to rotate their body frequently and extend their arms to manipulate the overlaid virtual contents, then rotational gain might prevent them from physically interfering with each other.

In our experimental setting, participants did not notice that they are being relocated, due to the subtlety of the translational gain. In future work, we would like to try to determine with more accuracy than our pilot studies, how much gain can be applied before users recognize the shifting, as Steinicke and colleagues [39] analyzed for redirected walking. The more translational gain we can apply without users noticing, the larger the distance we can shift. Thus, we can keep users even farther apart from each other, assuming we also increase $\boldsymbol{\tau}$, as well as increase the range of shifting.

Acknowledgements

We thank Sean White, Steven Henderson, and Christian Holz for their insightful thoughts and helpful feedback. This work was supported in part by NSF Grant IIS-01-21239, the Games for Learning Institute, and a generous gift from Microsoft Research.

References

[1] H. Asama, K. Ozaki, H. Itakura, A. Matsumoto, Y. Ishida, and I. Endo, Collision Avoidance among Multiple Mobile Robots Based on Rules and Communication, in *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, Osaka, Japan, 1991.

[2] I. Barakonyi, T. Fahmy, and D. Schmalstieg, Remote collaboration using Augmented Reality Videoconferencing, in *Proc. Graphics Interface 2004*, London, Ontario, Canada, pp. 89-96, 2004.

[3] I. Barakonyi, M. Weilguny, T. Psik, and D. Schmalstieg, Monkey-Bridge: autonomous agents in augmented reality games, in *Proc. ACM SIGCHI Int. Conf. on Advances in computer entertainment technology*, Valencia, Spain, pp. 172-175, 2005.

[4] S. Benford, A. Crabtree, M. Flintham, A. Drozd, R. Anastasi, M. Paxton, N. Tandavanitj, M. Adams, and J. Row-Farr, Can you see me now?, in *Trans. on Computer-Human Interaction*, vol. 13, no. 1, pp. 100-133, March 2006.

[5] M. Billinghurst and H. Kato, Collaborative augmented reality, in *Proc. Communications of the ACM*, pp. 64-70, 2002.

[6] S. Bjork, J. Falk, R. Hansson, and P. Ljungstrand, Pirates! Using the Physical World as a Game Board, in *Proc. Interact 2001*, Tokyo, Japan, 2001.

[7] W. Broll, J. Ohlenburg, I. Lindt, I. Herbst, and A.-K. Braun, Meeting technology challenges of pervasive augmented reality games, in *Proc. 5th ACM SIGCOMM workshop on Network and system support for games*, Singapore, p. 28, 2006.

[8] S. Bryson and C. Levit, The virtual windtunnel: an environment for the exploration of three-dimensional unsteady flows, in *Proc. IEEE Visualization*, pp. 17-24, 1991.

[9] C. Cai, C. Yang, Q. Zhu, and Y. Liang, Collision Avoidance in Multi-Robot Systems, in *Proc. IEEE Int. Conf. on Mechatronics and Automation*, Harbin, Heilongjiang, China, 2007.

[10] A. D. Cheok, K. H. Goh, W. Liu, F. Farbiz, S. W. Fong, S. L. Teo, Y. Li, and X. Yang, Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing, in *Personal and Ubiquitous Computing*, vol. 8, no. 2, pp. 71-81, May 2004.

[11] A. D. Cheok, A. Sreekumar, C. Lei, and L. N. Thang, Capture the Flag: Mixed-Reality Social Gaming with Smart Phones, in *Proc. ACM SIGCHI Int. Conf. on Advances in computer entertainment technology,* vol. 5, pp. 62-69, 2006.

[12] A. D. Cheok, X. Yang, Z. Z. Ying, M. Billinghurst, and H. Kato, Touch-Space: Mixed Reality Game Space Based on Ubiquitous, Tangible, and Social Computing, in *Personal and Ubiquitous Computing,* vol. 6, nos. 5-6, pp. 430-442, December 2002.

[13] N. Cooper, A. Keatley, M. Dahlquist, S. Mann, H. Slay, J. Zucco, R. Smith, and B. H. Thomas, Augmented Reality Chinese Checkers, in *Proc. ACM SIGCHI Int. Conf. on Advances in computer entertainment technology*, Singapore, pp. 117-126, 2004.

[14] M. Diaz, M. Alencastre-Miranda, L. Munoz-Gomez, and I. Rudomin, Multi-User Networked Interactive Augmented Reality Card Game, in *Proc. Int. Conf. on Cyberworlds*: IEEE Computer Society, 2006.

[15] M. Fiala, ARTag, a fiducial marker system using digital techniques, in *Proc. CVPR 2005,* pp. 590-596, 2005.

[16] R. C. Fitzpatrick, J. E. Butler, and B. L. Day, Resolving Head Rotation for Human Bipedalism, in *Current Biology,* vol. 16, no. 15, pp. 1509-1514, 8 August 2006.

[17] D. Fox, W. Burgard, S. Thrun, and A. B. Cremers, A Hybrid Collision Avoidance Method For Mobile Robots, in *Proc. IEEE Int. Conf. on Robotics & Automation*, pp. 1238-1243, 1998.

[18] M. Hakkarainen and C. Woodward, SymBall: camera driven table tennis for mobile phones, in *Proc. ACM SIGCHI Int. Conf. on Advances in computer entertainment technology*, Valencia, Spain, pp. 391-392, 2005.

[19] A. Henrysson, M. Billinghurst, and M. Ollila, AR tennis, in *Proc. ACM SIGGRAPH Emerging technologies*, Boston, Massachusetts, p. 1, 2006.

[20] A. Henrysson, M. Billinghurst, and M. Ollila, Face to Face Collaborative AR on Mobile Phones, in *Proc. 4th IEEE/ACM Int. Symp. on Mixed and Augmented Reality*: IEEE Computer Society, 2005.

[21] V. Interrante, B. Ries, and L. Anderson, Seven League Boots: A New Metaphor for Augmented Locomotion through Moderately Large Scale Immersive Virtual Environments, in *Proc. IEEE Symp. on 3D User Interfaces*, pp. 167-170, 2007.

[22] H. Kato and M. Billinghurst, Marker tracking and hmd calibration for a video-based augmented reality conferencing system, in *Proc. 2nd IEEE and ACM Int. Workshop on Augmented Reality '99* San Francisco, CA, pp. 85-94, 1999.

[23] K. Kiyokawa, M. Billinghurst, S. E. Hayes, A. Gupta, Y. Sannohe, and H. Kato, Communication Behaviors of Co-Located Users in Collaborative AR Interfaces, in *Proc. 1st Int. Symp. on Mixed and Augmented Reality*: IEEE Computer Society, 2002.

[24] D. C.-M. Leung, P.-S. Au, I. King, and E. H.-H. Yau, Remote augmented reality for multiple players over network, in *Proc. Int. Conf. on Advances in computer entertainment technology*, Salzburg, Austria , pp. 220-223, 2007.

[25] Lidgren, Lidgren Network Library, http://code.google.com/p/lidgren-network/.

[26] I. S. Mackenzie and S. Riddersma, Effects of output display and control-display gain on human performance in interactive systems, in *Behaviour & Information Technology*. vol. 13, no. 5, pp. 328-337, 1994.

[27] A. Mulloni, D. Wagner, and D. Schmalstieg, Mobility and social interaction as core gameplay elements in multi-player augmented reality, in *Proc. 3rd Int. Conf. on Digital Interactive Media in Entertainment and Arts* Athens, Greece: ACM, 2008.

[28] Newton Game Dynamics, http://www.newtondynamics.com/.

[29] T. Nilsen and J. Looser, Tankwar - Tabletop war gaming in augmented reality, in *Proc. PerGames 2005*, 2005.

[30] Nintendo, Wii, http://wii.nintendo.com/.

[31] O. Oda and S. Feiner, Goblin XNA, http://goblinxna.codeplex.com/.

[32] O. Oda, L. J. Lister, S. White, and S. Feiner, Developing an augmented reality racing game, in *Proc. INTETAIN '08*, Playa del Carmen, Mexico, 2008.

[33] T. Ohshima, K. Sato, H. Yamamoto, and H. Tamura, RV-Border Guards: A Multi-player Mixed Reality Entertainment, in *Trans. of the Virtual Reality Society of Japan*, pp. 699-705, 1999.

[34] T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura, AR^2Hockey: A case study of collaborative augmented reality, in *Proc. IEEE Virtual Reality Annual International Symposium*, pp. 268-275, 1998.

[35] V. Paelke, C. Reimann, and D. Stichling, Foot-based mobile interaction with games, in *Proc. ACM SIGCHI Int. Conf. on Advances in computer entertainment technology*, Singapore, pp. 321-324, 2004.

[36] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa, The go-go interaction technique: non-linear mapping for direct manipulation in VR, in *Proc. ACM Symp. on User interface software and technology*, pp. 79-80, 1996.

[37] S. Razzaque, Z. Kohn, and M. C. Whitton, Redirected Walking, in *Proc. Eurographics*, pp. 289-294, 2001.

[38] G. Reitmayr and D. Schmalstieg, Mobile Collaborative Augmented Reality, in *Proc. IEEE and ACM Int. Symp. on Augmented Reality (ISAR'01)*: IEEE Computer Society, 2001.

[39] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, Analyses of human sensitivity to redirected walking, in *Proc. 2008 ACM Symp. on Virtual reality software and technology*, Bordeaux, France, pp. 149-156, 2008.

[40] Z. Szalavari, E. Eckstein, and M. Gervautz, Collaborative gaming in augmented reality, in *Proc. ACM Symp. on Virtual reality software and technology* Taipei, Taiwan: ACM, 1998.

[41] Twister, http://www.hasbro.com/twister/default.cfm?page=Entertainment/History&src=endeca.

[42] D. Vogel and P. Baudisch, Shift: a technique for operating pen-based interfaces using touch, in *Proc. SIGCHI Conf. on Human factors in computing systems*, pp. 657-666, 2007.

[43] VTT, ALVAR, http://www.vtt.fi/multimedia.

[44] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg, Towards Massively Multi-User Augmented Reality on Handheld Devices, in *Proc. 3rd Int. Conf. on Pervasive Computing*, Munich, Germany, pp. 208-219, 2005.