LVC Interaction within a Mixed Reality Training System

Brian Pollock  
*Iowa State University*

Eliot H. Winer  
*Iowa State University, ewiner@iastate.edu*

Stephen B. Gilbert  
*Iowa State University, gilbert@iastate.edu*

Julio de la Cruz  
*United States Army Research Laboratory*

Follow this and additional works at: [http://lib.dr.iastate.edu/imse_conf](http://lib.dr.iastate.edu/imse_conf)  
Part of the [Operational Research Commons](http://lib.dr.iastate.edu/opsci), and the [Other Engineering Commons](http://lib.dr.iastate.edu/engg/)

---

**Recommended Citation**

[http://lib.dr.iastate.edu/imse_conf/53](http://lib.dr.iastate.edu/imse_conf/53)

This Conference Proceeding is brought to you for free and open access by the Industrial and Manufacturing Systems Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Industrial and Manufacturing Systems Engineering Conference Proceedings and Posters by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
ABSTRACT

This system was developed and tested in an immersive, reconfigurable, and mixed reality LVC training system for the dismounted warfighter at ISU, known as the Veldt, to overcome LVC interaction challenges and as a test bed for cutting-edge technology to meet future U.S. Army battlefield requirements. Trainees interact physically in the Veldt and virtually through commercial and developed game engines. Evaluation involving military trained personnel found this system to be effective, immersive, and useful for developing the critical decision-making skills necessary for the battlefield. Procedural terrain modeling, model-matching database techniques, and a central communication server process all live and virtual entity data from system components to create a cohesive virtual world across all distributed simulators and game engines in real-time. This system achieves rare LVC interaction within multiple physical and virtual immersive environments for training in real-time across many distributed systems.

Keywords: Mixed Reality, Military Simulation, LVC, Distributed Systems, Training

1. BACKGROUND

Physical training for the United States military provides a realistic environment for the warfighter, however these solutions are hindered by high costs and scenario inflexibility. Alternatively, virtual serious games offer high scenario flexibility and low cost, but seriously reduce scenario immersion and realism. Live, virtual and constructive training systems attempt to combine these two approaches into an effective, flexible and low cost solution to training.

1.1 Physical Training systems

Military operations in urban terrain (MOUT) sites are the typical locations for force-on-force physical training involving live opposing forces and simulated fire. Commonly the size of a town, MOUT sites are constructed of permanent materials and require personnel transportation to the site for training. More recently constructed MOUT sites also involve highly detailed sets, actors and pyrotechnics. These sets are populated with native speaking actors from the theatre of war moving about markets, cooking food, and engaging in other civilian behavior. Scripted hostile engagements prompt pyrotechnics and olfactory deployments to accustom the warfighter’s senses to the battlefield. Battlefield medical simulation can involve amputees with Hollywood grade makeup and actors donning “cut suits” which bleed, accommodate intravenous treatment, withstand sutras, and produce odors. The high fidelity of interaction for these exercises come at the cost of static city layouts, long scenario changeover, high overhead and large logistic.
cost. Decreasing armed service budgets and proven effectiveness of virtual training\(^2\) have shifted much focus to low cost simulations and serious games as a solution\(^3\).

1.2 Virtual Serious Games

Serious games commonly involve game engines and range in application from medical\(^4\) to combat situational\(^5\) to language and cultural\(^6\). These games exhibit a low cost, easily distributable solution requiring only seconds to switch between scenarios of different terrain, complexity and objective. After action review (AAR) modules can replay training scenarios and generate statistics to provide individual and group information over typical MOUT site capabilities.

A review of existing training applications reveals great diversity of game engines involved in military training. Often employing first-person-shooter (FPS) interfaces, these engines consist of both commercially and non-commercially developed systems including Virtual Battlespace 2 (VBS2), Delta3D, CryEngine, and Unity3D. This diversity indicates there is no single engine that best fulfills the vast array of training requirements for the warfighter. As these requirements increase in complexity, combining multiple game engines becomes necessary to incorporate the best features of each to meet these requirements.

LVC training systems integrate physical (live) trainees, (virtual) trainees controlling avatars, and artificially intelligent (constructive) avatars for interaction within one coherent environment. Often LVC training systems incorporate game engines for management and manipulation of the virtual aspects of that environment. Successfully combining multiple game engines into a single architecture would enable a training environment with the individual strengths of each engine. In addition, this system becomes an evolving platform that can be easily integrated with other simulations and upgraded as graphics, game engine, and simulation technology advance. Despite the advantages of multiple game engine systems, little published work exists on the development and implementation of such systems.

1.3 Mixed Reality Training Environments

Mixed reality environments for LVC training offer physical interaction while maintaining many advantages of serious games. These systems typically use a single game engine, but integrate additional components such as tracking systems, haptic devices, and mobile devices into the training experience. While multiple game engine systems are rare, a number of successful mixed reality LVC training environments for the dismounted warfighter have been developed.

FlatWorld\(^8\) is an example of a small, room-sized mixed reality environment, which uses rear screen projectors, sound, and props within a room to immerse the trainee. The main limitations with this small area environment are the restriction of physical movement, lack of adequate space for squad-based scenarios, limited virtual trainee interaction, and inability to quickly customize the environment for different training scenarios.

The Infantry Immersion Trainer (IIT) at Camp Pendleton is an example of a large environment for squad-based immersive training. The 32,000 sq.ft. IIT contains a layout of rooms and alleys populated with both live and constructive opposing forces and civilians. The IIT overcomes the space and squad limitations of the small training environment, however the live-constructive interaction is restricted without real-time tracking technology throughout the entire space. Indicative of the challenges when creating a LVC mixed reality training system, both FlatWorld and IIT still provide limited or absent ability for interaction between some LVC entities (i.e. physical-virtual, constructive-virtual, etc.).

1.4 Distributed Simulation Systems

Distributed simulation systems (DSS) attempt to combine multiple game engine systems and mixed reality environments. Distributed simulation systems have been developed for combining virtual agents, such as semi-automated forces (SAF), with a game engine\(^9\) and live tracked objects\(^10\).

Another DSS proposed for integration of many gaming and simulation components\(^11\) identified challenges when integrating heterogeneous software components. The architecture is a conceptually viable solution for mixed reality environments, but has not yet been implemented. Also, the proposed system is limited in applicability to an LVC system as it describes a purely virtual training system.
While DSS offers promise over multiple game engine systems or mixed reality environments alone, challenges remain. For example, communication is often one-way between system components, limiting true entity interaction among those components.

2. THE VELDT ENVIRONMENT

The training environments surveyed provide many unique capabilities, but none meet all of the requirements necessary for an adaptive, customizable LVC training system incorporating multiple game engines, stereo vision, and virtual and live tracked entities. In addition to these requirements, all the varying data feeds must be processed and transmitted quickly enough to enable real-time updates of the entire distributed system.

To provide a test bed for such a system, the Veldt was developed at Iowa State University (ISU) as a flexible mixed reality LVC training environment for the dismounted warfighter. This system combines the advantages of a physical environment where trainees can train through physical navigation and natural reaction with the flexibility of virtual training through integrated displays within a physical scene as shown in Figure 1.

In the physical space, replica weapons, helmets, and vests along with a reconfigurable set of walls outfitted with façades immerse the trainees in the mixed reality environment. Trainees, weapons, and props are tracked in real time by a uniquely configured infrared tracking system capable of millimeter precision. Trigger pulls from weapons are recorded via radio frequency (RF) through custom electronics. Real ballistics are not fired, but trajectories are calculated geometrically based on the position of the weapon at time of trigger pull to enable virtual ballistics to be displayed on the integrated displays as well as other networked game engines.

The physical set was constructed to be modular through “L”-shaped wall segments. These segments feature interchangeable texture facades (e.g., brick, stone, and plaster) to allow quick customization for different training scenarios. For example, the Veldt can be reconfigured from a checkpoint scenario in an open street with industrial style buildings to a close quarters tactical raid scenario in a residential marketplace in less than 30 minutes.

For the reconfigurability of the virtual components of this mixed reality environment, multiple stereoscopic displays are utilized throughout the Veldt. These integrated displays range in size and projection technology to fit in windows, alleys, doorways, roads, and rooms. This allows views to extend far beyond the physical limitations of the room in which the Veldt is housed. In addition, this configuration allows multiple virtual and constructive entities to interact with live participants in various training scenarios.
3. LVC MIXED REALITY INTERACTION

3.1 DeltaJug

Live trainee interaction occurs through multiple distributed displays and simulated weapon fire within the Veldt. While screen modularity allows quick reconfiguration between various scenarios, such as a checkpoint or network of rooms, it also required the development of a clusterizable game engine DeltaJug for synchronized stereo visualizations of the virtual world on multiple displays.

The implementation of multiple displays within a training environment is necessary to increase environment flexibility and blend both virtual and real worlds seamlessly. This provides enhanced immersion over “shooting gallery” type single display systems. Multiple display systems also place unique restrictions upon the game engine driving the system. For a synchronous scene among the displays, each display must render the exact same frame at the exact same moment, albeit through a different view frustum. Swap locking of slave nodes by a master node is controlled through transmission control protocol (TCP) communication, which virtually eliminates latency between each display. This clustered graphics approach is critical to allowing the scene to be continuous on the various disjointed display surfaces, but adds additional network complexity. Most commercial and open source game engines do not allow scalability for clustered graphics beyond a few nodes of a computer system. Thus, one had to be designed for the many nodes required, sometimes as many as 96.

The open source game engine Delta3D\textsuperscript{12} provided a platform for military training\textsuperscript{13,14} with low-level source code access that could be altered to provide the clustered graphics capabilities required. Delta3D was combined with VR Juggler, a networking and hardware abstraction API commonly used for clustered graphics application development. This clusterizable game engine was termed DeltaJug\textsuperscript{15}.

As an open source game engine based upon a virtual reality framework, DeltaJug can readily integrate multiple tracking systems and hardware components associated with the physical environment. These components are added to a DeltaJug application through XML-based configuration files, thereby eliminating specific code changes when switching hardware systems. This allows the same application code to be used on a single wall monoscopic system with sonic tracking and a multisided, stereoscopic, immersive virtual reality system with optical tracking with only a change in configuration file.

In the Veldt, DeltaJug receives all information about physical trainees and objects from the tracking system and RF signals from weapon trigger electronics; therefore it is responsible for the creation and management of all live entities in the virtual world. DeltaJug creates virtual representations of these live entities on initialization and updates them each frame with information supplied from the tracking system and weapon triggers. DeltaJug applies these entity updates across the graphics node cluster running the various displays in the Veldt for synchronous entity updates every frame.

3.2 Virtual Battlespace 2

DeltaJug solved a major implementation issue of synchronization, however it does not contain an extensive model library, allow quick and easy scenario authoring, or contain a polished after action review tool. Therefore, a commercial game engine common to U.S. military training, Virtual Battlespace 2 (VBS2), was utilized to employ these features. While VBS2 was determined most appropriate for the Veldt implementation, other common game engines such as Unity3D or CryEngine could have been implemented with similarly ease.

VBS2’s scenario authoring ability enables trainers to easily create scenarios and assign behaviors for virtual and constructive entities. Trainees can virtually interact with the system through VBS2’s FPS style interface, Figure 2, and multiple virtual trainees can interact from multiple instances of the game engine through VBS2’s networked mission capability. Trainers can additionally participate in the scenario as observers and record the mission along with virtual and live entity performance metrics for AAR.

3.3 C6 Collaboration

Interservice capability is offered within the C6, a high-resolution six sided cave automatic virtual environment (CAVE),
running Battlespace\textsuperscript{17,18}, a command and control application for semi-autonomous unmanned aerial vehicles (UAVs). The Battlespace application was developed at Iowa State University using OpenSceneGraph to enable one operator to control a large area through simultaneous control of multiple UAVs through semi-autonomous path planning. In addition to UAV planning, Battlespace provides a commander perspective of the entire ground scenario, shown in Figure 3.

Figure 2: VBS2 FPS interface

Figure 3: C6 Battlespace application in command and control exercise

4. ARCHITECTURE

This modular design localizes control of entity virtual representation to their respective applications. For synchronizing virtual worlds between distributed systems, a central communication server was created to disperse state updates to all applications connected to the distributed system.

3.1 Central Communication Server

To create a true DSS between the Veldt and multiple game engines, a central communication server was designed to connect all system components using various communication protocols. The server accepts communication from all components in the system and converts their data from the sending component’s protocol into a world state within the communication server. After conversion, the communication server identifies which components have requested particular information and the data is converted from the world state into a receiving component’s local protocol for transmission. The communication server was designed to accommodate any communication protocol from many components and to distribute data as quickly as possible to minimize latency. This process is similar for all components connected to the communication server and this architecture can be extended to include live vehicles and mobile devices as well as other game engines and simulations.

With all communication between major components managed by a central communication server, shown in Figure 4, the addition of game engines or simulators need to communicate only with this server, not all other components. Additional components require registration and possibly a new method converter in the communication server. As a result, this central communication server reduces integration of components for a DSS, multiple game engine, and mixed reality LVC training environment.
Numerous challenges were overcome throughout the creation of this flexible LVC training system. The primary challenges of a multiple game engine system involve scene synchronization, terrain generation, and real-time tracking in a reconfigurable environment.

3.1 Distributed Simulation Issues

LVC training systems contain similar issues as other distributed software architectures. Regardless of system specifics, these systems are known to have network latency, hardware architecture differences, software system delays, and other potential challenges. Previous research has also identified protocol communication, such as DIS, between simulations to have high network bandwidth costs\(^{19}\), however technical advancements since have largely overcome these challenges.

Network latency between simulations can result in different states among components. For example, high network latency would cause VBS2 controlled actors represented within DeltaJug to not properly synchronize across the system, affecting training. Computational methods exist for correcting for this unavoidable latency\(^{20}\), however the best approach is to minimize this issue locally through system design.

To test system latency, many components spread between two buildings were connected in an interservice LVC exercise at Iowa State University. This exercise involved one master and two slave DeltaJug nodes, the Veldt tracking system, and two VBS2 instances in the Veldt and command and control room, Battlespace in the C6, and an additional tracking system external to the Veldt and C6. Both Veldt and UAV simulator computer clusters’ nodes were synchronized through the same swap locking mechanism present within VRJuggler. Network speed, software delays, and hardware differences were not identified as an issue for latency. This exercise occurred in real-time with multiple entity updates a second for all components, and all component information transported through the communication server.

3.2 Scene Synchronization

Scene synchronization of both game engines and the physical environment is necessary for a cohesive view of the scenario regardless of live or virtual interaction. The first challenge for scene synchronization is coordinate conversion among various components. Each tracking system, game engine, and simulation could potentially have its own unique origin and coordinate system. The conversion of entity information for each component is vital to proper representation of entities within game engines. Scene synchronization also involves model database consistency across each game.
engine for collaboration between the trainees of those engines. When two game engines are visualizing a vehicle entity, it cannot resemble a white pickup truck in one game engine and a red dump truck in another game engine for proper communication between live and virtual trainees or AAR. This mismatch of models associated with an entity is often a result of improper communication configuration and model database inconsistencies between engines.

Many commercial game engines contain extensive model databases, but these are typically proprietary. Model databases available for open source game engines are more limited. For a multiple game engine system, a consistent model database must be configured for the entire system. This model database can be created or purchased if the model database is flexible enough to export to a variety of proprietary and open source formats from source, but this exportation approach has several drawbacks. The entire model database must be exported for each game engine added to the system utilizing an alternative model format and it is not guaranteed models can be exported proprietary formats, as may be required by some commercial game engines. An exportation approach assures identical models across the system, but incurs large costs in time, resources, and flexibility.

An alternate model-matching approach, implemented in the Veldt system, required less time and resources, utilizing the local model databases within open source and commercial engines. For this approach, a set of common model entities among the databases was identified. For example, a civilian pickup truck, an insurgent armed with an AK-74, and a desert HMMWV. Although the models might vary in appearance amongst the different engines, their native configuration makes entity creation and control much easier within those engines. Minimal visual differences were found to exist between local model databases because often models were created to represent the same physical object. Supplemental configuration of entity type identifiers may also be necessary within each game engine. Addition of an alternative game engine would only require configuration of protocol identifier information within either the communication server or within the added game engine to match the model database structure.

This model-matching approach provides a consistent database among components using existing models within each local database and can quickly add new components to the system through minimal configuration for each new database. Model-matching can result in a less extensive and non-exact database compared to the model exportation approach, however matching is more flexible when adding components. Model-matching also requires less time and expense than a model exportation solution, which requires exportation of the model database in each game engine’s format.

3.3 Real-Time Tracking of a Reconfigurable Environment

Another difficult challenge involved the tracking of physical objects and entities within the Veldt for representation in virtual environments. Many optical and infrared tracking systems are designed to track in an open environment. However, the Veldt or any other physical immersive environment contains many obstructions to line of sight tracking cameras such as barriers, doors, walls, and people. Initial attempts at designing a tracking system to account for these additional challenges were to involve 24 cameras for tracking in the Veldt space, more than typically used in an open environment, with specific positions and orientations. For example, to track a soldier crouched in a narrow hallway, at least three cameras needed line-of-sight to tracking markers on the soldier. To define the positioning of the cameras, a number of Veldt wall configurations were drafted and tracking areas of high importance were identified for each scenario. Lastly, it was understood that some tracking cameras might need relocation for some configurations. With the inclusion of more cameras this relocation could be eliminated, however with additional cost.

6. EVALUATION

The Research Institute for Studies in Education (RISE) at ISU performed an initial evaluation of this system. In this evaluation twelve participants from the ISU Reserve Officer Training Corps (ROTC) took part in a single training scenario within the Veldt. These participants took part in a room-clearing scenario in teams of two involving live, virtual, and constructive opposing forces.

First a team was oriented to the environment. Teams put on their helmets and practiced firing their replica weapons. Next, the team was given a briefing of the scenario identifying what forces were believed to be located within the Veldt environment (e.g. number of live and virtual warfighters and insurgents). After orientation and briefing, each team took part in the simulation, completing the exercise twice.
Immediately upon completion of both runs of the scenario, each team was taken to a separate room for evaluation by RISE staff. Participants completed a web-based survey individually and teammates were verbally asked questions by RISE staff on their experiences within the Veldt. In addition, video recording and after action review provided location, orientation, weapon use, and accuracy information for each participant. The survey drew heavily from the Presence Questionnaire developed by Witmer and Singer\textsuperscript{21}. Theory for development of the survey followed work on the factor structure of the presence questionnaire\textsuperscript{22}.

5.1 Web Survey Results

The web survey yielded three main findings: the Veldt is an effective system for training, the mixed reality environment was visually immersive and engaging, and sound is essential to improve immersion.

When asked how their ability to meet training objectives had changed based on their Veldt experience, 84% of participants reported \textit{Moderate or Vast improvement} in their ability to engage enemy combatants. 75% reported \textit{Moderate or Vast improvement} in their ability to make quick decisions in a stressful environment. Teams also demonstrated a noticeable increase in completion speed of the scenario their second time.

All participants (100\%) agreed (indicated either \textit{Agree somewhat} or \textit{Agree completely}) that the visual aspects of the Veldt environment involved them. While 83\% agreed that their senses were completely engaged, 84\% agreed that they felt involved in the virtual environment and 67\% agreed that their interactions in the Veldt environment seemed natural or true to life.

Finally, participants reported that absence of sound appeared to diminish immersion with most participants (67\%) rated the ability to locate enemy fire by sound as \textit{Poor} or \textit{Very Poor}. However, most participants (58\%) could locate the source of enemy fire visually with responses of \textit{Good} or \textit{Very Good}.

5.2 Team Interview Responses

Interviews with the teams by RISE staff provided responses indicating participants felt that engaging in these activities prior to entering combat situations would increase their understanding of decision-making, especially as it pertained to engaging with noncombatants. One participant remarked, tracking provides useful information in terms of a trainee’s accuracy in shooting and can determine if they follow procedures. This reinforces the advantages of real-time tracking that offers trainers more data and trainees another level of feedback.

In congruence with survey responses, participants provided several suggestions on ways to improve the Veldt environment including adding more “friction” or gunfire, obstacles, ambient noise, and ambiguous directions. While participants were generally positive on the visual immersion of the Veldt environment they indicated they would welcome greater physical details such as flooring, ceilings, furniture, and holes in walls to enhance the realism of the environment.

7. DISCUSSION

The developed mixed reality LVC training system provided an immersive, flexible environment more engaging than computer based training with less cost than live training exercises. This system included live entity tracking, multiple game engines, virtual & constructive entities, replica weapons & apparel and high-end graphic simulations. This framework utilizes the best features of multiple game engines, creating a system that can evolve with technological advances, and lessens the integration challenges of interservice LVC training. The Veldt was the first test bed of such a system that met all of the requirements necessary for a multiple game engine, mixed reality, DSS LVC training system for the dismounted warfighter.

Throughout development, precise tracking of people and weapons within a complex obstructed environment remains difficult for tracking technology. Despite this challenge, visualization of tracked entities within two training scenarios and one interservice exercise occurred in real-time across all game engines with accurate location, orientation, and
posture. Future work involving advanced tracking of entities through motion capture for skeletal models of live entities would improve the virtual realism of these entities. Further work could also examine to what extent this information could be provided to commercial game engines through network communication without source modification.

Each graphics display viewpoint in the Veldt is statically determined according to scenario layout. Tracking system data provides adequate information to render views from participants’ head locations however both software and hardware solutions for multiple head tracking would reduce immersion within the mixed reality environment and are often not easily scalable. Future work will investigate the perception issues driving head tracking solutions in order to implement a multiple head tracking solution.

All study participants found the Veldt visually engaging, 83% felt their senses were completely engaged and a majority felt their interaction as true to life. While these study results are encouraging, improved tracking, implementation of spatial sound and integration of tactile feedback will be investigated to enhance immersion and interaction. For example, future work on tactile feedback could provide physical indication of injury and potentially act as a nonverbal intelligent tutoring indicator.

Technological improvements to game engines for clusterizable situations and openness of commercial code for simulation will improve the LVC training situation. However, a uniform game engine across the military training and simulation field is not foreseen in the near future. As training environments become more complex and distributed, connecting the various hardware and software components will become more difficult. Continued trends of standalone solutions will further fragment these complex training systems when integration is necessary for interservice LVC training. This research offers a solution to this problem by providing a framework to integrate just about any live, virtual, or constructive training system.

The created multiple game engine, mixed reality, DSS LVC training system eliminates standalone dependency on specific components such as game engines and tracking systems. Without this dependency, training systems can combine the benefits of different technologies for a superior system customizable to a system’s training requirements. This research has proven the ability of such mixed reality LVC training systems to offer effective training, immersive interaction and scenario flexibility. Additionally, the challenges and areas for improvement identified in the Veldt will help guide future attempts to create low cost, mixed reality, LVC training solutions for the dismounted warfighter.

ACKNOWLEDGEMENT

The authors would like to acknowledge RDECOM - STTC for their funding and support of much of the work included.

REFERENCES


