AR-Weapon: Live Augmented Reality based First-Person Shooting System

Zhiwei Zhu, Vlad Branzoi, Mikhail Sizintsev, Nicholas Vitovitch, Taragay Oskiper, Ryan Villamil, Ali Chaudhry, Supun Samarasekera and Rakesh Kumar

SRI International, Princeton, NJ,08540

{zhiwei.zhu,vlad.branzoi,mikhail.sizintsev,rakesh.kumar}@sri.com

Abstract

This paper introduces a user-worn Augmented Reality (AR) based first-person weapon shooting system (AR-Weapon), suitable for both training and gaming. Different from existing AR-based first-person shooting systems, AR-Weapon does not use fiducial markers placed in the scene for tracking. Instead it uses natural scene features observed by the tracking camera from the live view of the world. The AR-Weapon system estimates 6-degrees of freedom orientation and location of the weapon and of the user operating it, thus allowing the weapon to fire simulated projectiles for both direct fire and non-line of sight during live runs. In addition, stereo cameras are used to compute depth and provide dynamic occlusion reasoning. Using the 6-DOF head and weapon tracking, dynamic occlusion reasoning and a terrain model of the environment, the fully virtual projectiles and synthetic avatars are displayed on the user's head mounted Optical-See-Through (OST) display overlaid over the live view of the real world. Since the projectiles, weapon characteristics and virtual enemy combatants are all simulated they can easily be changed to vary scenarios, new projectile types and future weapons. In this paper, we present the technical algorithms, system design and experiment results for a prototype AR-Weapon system.

1. Introduction

Augmented Reality enhances and augments a person's experience of the real world. We have built a prototype AR-based first-person shooting system (Figure 1), which allows the users to physically move through the real environment during training or game-playing. Virtual characters and effects are inserted in the user's view of the live real world through head-mounted Optical See-Through (OST) displays such that the content exactly matches and blends with the real world. This is achieved by very precisely tracking the individual user's head location and orientation in the world by user-worn sensors. A back-end game-engine generates reactive AR content based on user's movements and their location/orientation.

Figure 1: The AR-Weapon prototype system.

In addition, each weapon is precisely tracked relative to the real world as well with respect to the user's helmet using a weapon mounted sensor package. Therefore, through the OST HMD, the user is able to see and aim at real or virtual targets as he normally would do when using his unaided eye. Note, In order for the inserted objects not to jitter or drift as the user moves around, the estimated 6-DOF pose must be very accurate and have low latency.

Existing shooting-related systems have a limited ability to track users during games and to adapt virtual actions to the movements of the users. Recently, in military applications, a few Mixed Reality systems such as the Infantry Immersive Trainer [2] and the Automatic Performance Evaluation and Lessons Learnt (APELL) system [7] have been deployed at Camp Pendleton and other Marine Corp's MOUTs (Military Operations on Urban Terrain). These systems use video projectors to project images of virtual actors on walls of rooms within a training facility. These systems are limited to indoor exercises and require significant infrastructure.

There are very few systems which can track users both indoors and outdoors. GPS based systems may be used for providing location outdoors. However, the performance of these outdoor-only systems decreases in challenging GPS limited situations. Ultra-wideband (UWB) based systems have been used for indoor tracking of users to foot (30 cm) level accuracies [4] but do not provide orientation information. Finally, none of these systems meet the challenging requirement for AR [1, 5] where both location and orientation of the user's head must be tracked to cm level accuracy for location, and less than 0.05 degree accuracy for orientation. Overall, providing high accuracy

tracking over large indoor and outdoor areas (multiple square miles) is a very challenging problem.

The first real AR-based shooting game developed was ARQuake [12]. It is able to track the user's six degrees of freedom location and orientation so that the user is able to move freely in the physical world. However, there are quite a few issues with this prototype system. First, the system relies on GPS, compass and fiducial markers for tracking the navigation state and the robustness, accuracy and precision of the estimation is a big challenge. Second, during the firing, the aiming of the weapons is essentially the direction of the user's head, which is another major disadvantage of the system. Third, video-see-through glasses are used to display the views in front of the user captured by the cameras. The latency and resolution of these displays affect the user experience significantly. On the other hand, AR-Shooter [6] improves the usability of the weapon by adding a tracking camera on the gun. It developed an independent DSP-based hardware module to reduce the CPU's burden. However, the tracking was done using infrared markers, and no tracking device is used to estimate the user's head pose.

In order to overcome the shortcomings of the existing AR-based shooting systems, in this paper, we have developed a user-worn AR-Weapon shooting system that estimates precise 6-DOF orientation and location of the weapon and of the head of the user wearing the AR-system. Using this 6-DOF tracking, occlusion reasoning and a terrain model of the environment, virtual projectiles and their blast effects and simulated damage are displayed on the user's optical see-through eye-wear. This allows the user to aim and fire the AR-weapon to shoot simulated projectiles at synthetic avatars and objects during live runs. Since the projectiles, weapon characteristics and gaming enemy combatants are all simulated, they can easily be changed to vary scenarios, new projectile types and future weapons. In addition, dynamic stereo-based occlusion reasoning is implemented to give more realistic blending of the virtual insertions with the real scenes to improve the user experience. Details of the system description, technical algorithms and experimental results of the proposed prototype AR-Weapon system are described in the subsequent sections.

2. Overall Approach

Figure 2 illustrates a high level diagram of the developed AR-Weapon training system. It consists of a helmet-worn sensor and OST display package, an AR-enabled weapon and a light-weight processor mounted on the user's vest.

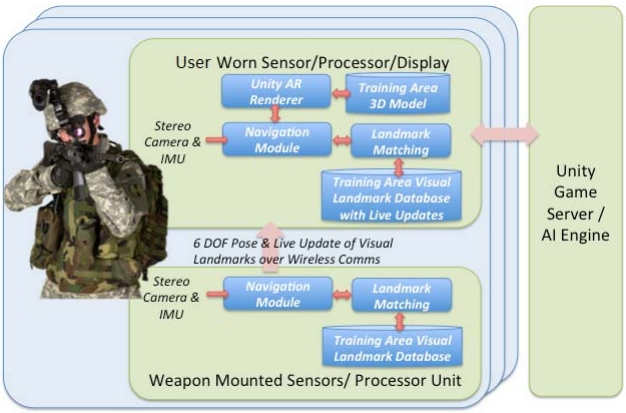

Figure 2: System architecture with integrated weapon tracking.

Specifically, the weapon instrumentation includes stereo cameras; a MEMs based IMU and a mobile processor unit running Linux OS. Live video and IMU data are ingested into the navigation module running on the mobile processor unit. The navigation module uses a landmark matching module to establish correspondences between features extracted from the live video and a pre-built Visual Landmark database of the training site. These correspondences provide the navigation module geo-constraints with respect to the training site. Navigation module fuses the landmark constraints when available, with continuous stereo based image feature tracks and IMU data to continuously estimate the weapon pose (position and orientation in 6 DOF). These poses and the landmarks are transmitted wirelessly to the user worn computer.

On the user worn computer we process data from the helmet worn sensors and visual-landmark matching module to establish 6 DOF pose (position and orientation) of the user head. The poses and landmarks from the weapon unit are also used to update the pre-built landmark database on the user worn system. This ensures that the user head pose is not only consistent with the global coordinates system of the prebuilt landmark database, but also consistent with the weapon's local coordinates system.

2.1. Navigation Module

Figure 3 shows the flow chart of the localization module. Central to the localization solution is an IMU-centric extended Kalman filter (EKF). The module implements an error state Extended Kalman filter that uses the 3-DOF accelerometer and Gyro to derive an IMU mechanization state and evaluates "Error-States" of other sensors (e.g., video, RF ranging) with respect to this mechanization. Video-based reasoning provides the high-fidelity localization to our solution. A real-time module

for doing visual odometry provides high-quality relative pose inputs. A visual landmark matching module enables longer range drift (location and orientation) corrections.

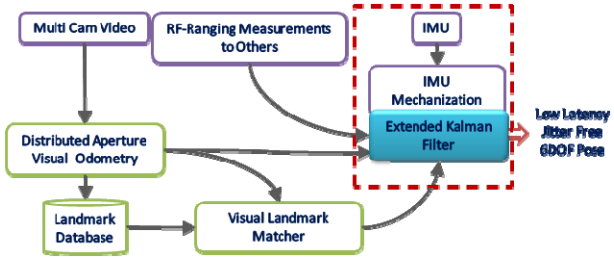

Figure 3: The flowchart of the navigation filter.

The localization module implements an IMU-centric error-state EKF approach [3,9] to fuse IMU measurements with external sensor measurements that can be local (relative), such as those provided by visual odometry, or global, such as those provided by landmark matching. This filter replaces the system dynamics with a motion model derived from the IMU mechanism. The filter dynamics follows from the IMU error propagation equations, which evolve smoothly and therefore are more amenable to linearization. This allows for better handling of the uncertainty propagation through the whole system. The measurements to the filter consist of the differences between the inertial navigation solution as obtained by solving the IMU mechanization equations and the external source data. The final filter estimate can automatically remove spurious measurements from external sensors, such as visual odometry when vision fails.

Global measurements are provided by matching the current image to a landmark database. Given a query image, landmark matching returns the found landmark shot from the database. This match is used to establish 2D to 3D point correspondences between the query image features and the 3D world model. The 2D-3D correspondences are applied as measurement equations in the error-states of the EKF filter. The landmark database is built offline and is matched in real-time to establish the global constraints online [8].

3. Distributed Visual Landmark Matching for Accurate Weapon Pointing

As shown in Figure 4, for the AR-Weapon system, it requires both the weapon-mounted navigation system and the user-worn navigation system to be precisely localized in a common coordinates system that is aligned with a pre-built landmark database of the training site.

Figure 4: The weapon and the user head align precisely together with the landmark database of the training area.

Landmarks database consists of visual features extracted from sparse collection of pre-collected video of the training site. The features extracted from the images are histograms of the oriented gradients (HOG). Because stereo cameras are utilized, the extracted features are tagged by their 3D location and are scaled by distance. Therefore, the framework proposed in [8] was used to match images across largely varying perspectives. When the landmark database is built, global bundle adjustment methods are used to make the 3D poses of these features consistent. In addition, the landmark database and the virtual 3D model used by the game-engine/renderer have to be aligned into the same coordinate system.

Figure 5 shows the workflow of steps that enable robust 6-DOF pose estimation of the weapon and helmet views.

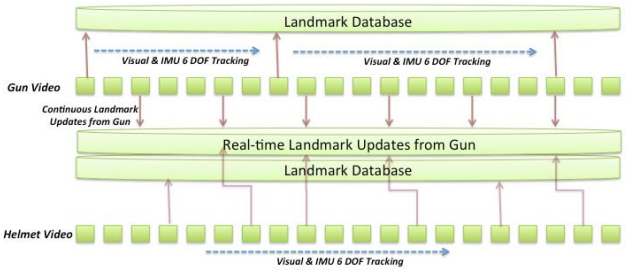

Figure 5: The work flow of weapon and helmet tracking enabling continuous 6-DOF global positioning.

The weapon system uses stereo video and IMU as input. Given the need to operate indoors we do not use GPS. It uses the visual landmark matching described above to match the live gun video to the pre-built landmark database. The stereo provides 3D constraints to match against the 3D landmark database to establish an initial pose relative to the training site. The large variety of gun positions and the sparseness of the landmark database limits how often we find landmark matches. In between

landmark matches, the Kalman filter uses the visual-odometry and IMU readings to perform 6-DOF dead-reckoning. This allows us to continuously track the weapon position and orientation in real-time.

We use the same process to track the user head movement using stereo and IMU sensors mounted on the helmet. This provides head-orientation and position with respect to the same landmark database independent of the gun-pose estimation. However, in certain cases, for instance areas of sparse landmarks or when the weapon obscures the helmet view, the estimate of the helmet pose can become inconsistent with respect to the estimated gun pose. To overcome this we continuously generate new visual-landmarks from the weapon navigation system and transmit these wirelessly to the helmet system as an additional source of landmarks for the helmet navigation system. This significantly increases the frequency of landmark matches on the helmet sensor and ensures the weapon pose and the head pose stay consistent. For engagement of the virtual target this relative accuracy is critical. In the "Experimental Results" section, we show additional details of the operational prototype system and its performance.

4. Unity AR Renderer with Occlusions

The AR rendering system is responsible for inserting virtual entities in real-world. The renderer is based on the commercial Unity 3D game engine and is tailored to support augmented reality use. The avatar control system for our simulation consists of a central simulation server and distributed rendering clients running on each of the trainee worn processors. The central server controls all avatars, and is responsible for synchronizing avatar states (such as position, orientation, and current animation) amongst all renderer clients. All trainee locations are conveyed to the central server, providing multi-participant support.

The insertion of avatars is based on reasoning of static and dynamic occlusions. Static occlusion is generated from pre-built training area model that has been imported into the Unity framework. The dynamic occlusion is generated by computing a depth map from the head-worn stereo cameras in real time.

Dynamic depth occlusion is particularly challenging due to strict requirements on fast processing, crisp 3D boundaries and maximum high density of results. In the current setup we rely on the hierarchical stereo algorithm originally proposed in [10] since it is fast and able to produce dense disparity maps with high quality 3D discontinuities. Furthermore, a CUDA implementation tailored to Augmented Reality application has been developed in [11], but it assumes horizontal rectified input and Nvidia processing hardware. To make the proposed system more practical, we need it to operate on small form factor gear, preferably using only multicore processor. Thus, the method [11] has been redesigned for multi-threaded CPU architecture to operate on vertical stereo input, which is transposed and processed as a standard stereo with horizontal epipolar lines because it allows for better cache coherence. Furthermore, the input images can be resized to keep up with real time processing requirements. Figure 6 depicts a stereo processing example for the current system described in details in later sections. In particular, 320x240 disparity maps are computed at average 20 fps rate using a Core-i7 based wearable PC that also performs head tracking and scene rendering.

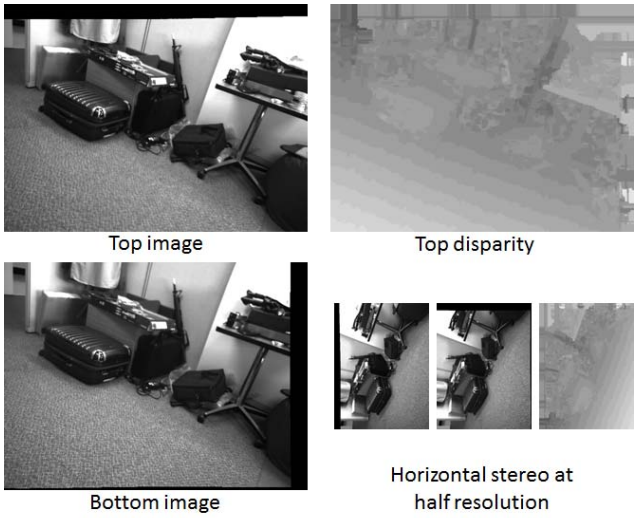

Figure 6: Real time stereo processing example. Vertical stereo feed is transposed as a horizontal stereo pair.

Since our AR setup is based on optical see-through display, the processed disparity map is converted to the depth map and rendered from the eye-piece perspective using the pose predicted from tracking module. Consequently, dynamic occlusion from estimated depth map is most effective for those objects that are part of the scene and non-moving, for which pose prediction of the user's head is sufficient. For dynamic moving objects, their individual pose also needs to be tracked and predicted for best results.

The 6 DOF pose computed from the Navigation module is used to generate the exact perspective of the trainees view for rendering. The navigation module uses the IMU combined with the visual processing to generate these poses at very high rate (100-200Hz). This ensures the rendered view as minimal latency relative the actual movement of the trainee. The renderer communicates with

a backend game-engine to generate entity behaviors. This allows rendering between multiple trainees to be coordinated and consistent.

For effects related to the weapon firing, the rendering engine uses the 6 DOF gun pose that is consistent with the world model to find intersections with the fused depth map to determine impact. The intersection can be with a virtual avatar or object in the system. In such case it can determine where the avatar/object is hit and generate appropriate reactive behaviors.

If the intersection is with the static training area model such as a wall we can show weapons effects on the wall. Furthermore, based on properties of the walls that are defined in the world model we can determine if the bullet can pass through the structure and impact a character behind the wall. If the intersection is with a dynamic object corresponds to a live player in the system and can provide feedback to the player through the game server.

5. System Setup

5.1. AR-Weapon Prototype Hardware

Figure 7 (left) shows our customized sensor-package mounted on the weapon. The sensor-rig consists of one pair of stereo-cameras (Point Grey Chameleon CMLN-13S2M-CS), one Inertial Measurement Unit (IMU) (Microstrain 3DM-GX3-25) and one HardKernel ODROID-XU processing board. All of them are packaged inside a compact enclosure that can be easily mounted onto the barrel of the weapon as shown in Figure 7**.**

Figure 7: The customized weapon sensor package.

Figure 7 (right) shows the Samsung mobile processor based board (ODROID-XU). It is equipped with Exynos5 Octa Cortex-A15 1.6GHZ quad-core and Cortex-A7 quad-core CPUs, PowerVR SGX544MP3 GPU and 2Gbyte LPDDR3 RAM. It has small dimensions with only 9.4cmx7.0cmx1.8cm, which allows us to package it into a compact and low-weight enclosure as shown in Figure 7 (left bottom). The stereo-cameras are able to run at 15fps with 640x480 pixel resolution on the ODROID-XU mobile processor board. Together with IMU unit running at 100Hz, the sensor-rig is able to estimate the 6-DOF pose of the weapon at 15fps robustly.

5.2. User Helmet-Worn System

Figure 8 shows our customized human wearable sensor rig package. It consists of a Cybermind Cyber-I Bi-nocular OST HMD, one pair of stereo-cameras (Point Grey Flea3 GigE FL3-GE-13S2C) and one IMU (Microstrain 3DM-GX3-25). The stereo cameras are arranged vertically for minimal intrusion to the user, and the images are captured at 15fps with 640x480 pixel resolution. The IMU unit operates at 100HZ and it is synchronized with the stereo cameras to form a multi-sensor navigation system to provide precise pose estimation. The sensor-rig and the HMD are rigidly mounted together, and their spatial relationship is calibrated in advance. Once the calibration is done, it is used to transform the pose estimated by the navigation system in order to accurately insert the synthetic objects in the HMD. Through the OST HMD, the user is able to see and aim at the real or virtual targets as he normally would do through his unaided eye.

Figure 8: The human wearable head-mounted sensor package with the Optical-See-Through display.

5.3. Low Latency Forward Prediction

For OST AR, accuracy of the pose estimates alone is not sufficient for delivering an acceptable user experience to the person who is wearing the HMD. For example, besides rendering a virtual marker at the correct location, the rendered marker also needs to appear with very little delay on the display. This is due to the fact that, in the OST framework, the user sees the real work as it is (not an image of it) and hence the equivalent "frame-rate" is essentially very high and there is no-delay in visual perception of the real world. Therefore, the associated rendered markers have to satisfy this highly demanding requirement in order for them to appear jitter-free when they are displayed. Otherwise as the user's head moves, the markers will appear to bounce around in the display since they will be lagging in time.

Video frames in general arrive (15 Hz in our case) at a much slower rate than the IMU samples (100 Hz in our case.) The pose estimates that incorporate the video frame information are generally available after a 40-50 msec processing delay. The pose requests from the renderer arrive asynchronously at the highest rate the renderer can

accommodate. After the renderer receives a pose it is displayed on the OST display after a certain amount of delay which is affected by both the display hardware latency and lag caused by the inefficiencies in the rendering pipeline and video graphics card. Figure 9 summarizes all latency sources along the pipeline in the system, and we have to carefully measure them and then compensate for them.

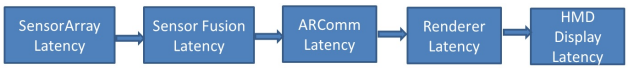

Figure 9: potential latency source along the pipeline.

The latency caused by the first two blocks in Figure 9 is compensated by using the IMU data that has been buffered in the system between the latest frame time and the current render time. Propagating the most recent pose by integrating all the IMU readings until the current time instant provides the most accurate no-latency camera pose which is sent to the renderer. In order to compensate for the remaining latencies in the system, a forward prediction mechanism is utilized to estimate the camera pose corresponding to a certain timestamp into the future given all the information that is available up until the render request. For this purpose, forward prediction performs a second-order extrapolation of the orientation using a window of past camera poses with the Kalman Filter [9]. For our AR-weapon system, we found that forward predicting 20ms ahead compensates the latency well.

6. Experiment Results

In order to evaluate the performance of the developed AR-Weapon system, a set of experiments were conducted in a long straight hallway (around 100 meters long). As shown in Figure 10, a printed bull's eye board and a LED light source were mounted on the wall at the end of hallway as targets. We moved the AR-weapon system back every 5 meters on the hallway and stopped for aiming at the targets. At each stop, we tried to aim the weapon exactly at the center of the target. For the distance less than 20 meters, we chose the bull's eye as the aiming target; for the distance larger than 20 meters, we chose the LED light source as the aiming target.

During the experiments, the AR-weapon system was running live. The image center of the camera mounted on the weapon was displayed as a Red Cross as shown in the left image of Figure 11. The scene point that the Red Cross pointed at was chosen as the aiming target of the weapon.

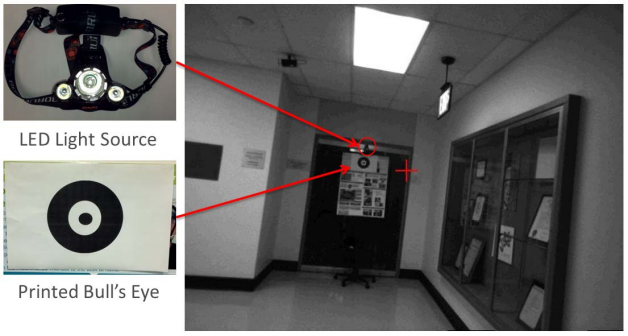

Figure 10: The setup of aiming targets.

The estimated 3D line of the weapon was displayed in the head-mounted camera as a virtual Red Line as shown in the right image of Figure 11. The scene point that the estimated 3D line of the weapon intersected with was the aimed target. The scene point that the Red Cross overlays on weapon image and the scene point that the estimated Red Line intersects with on helmet image will be the same point if the AR-weapon and AR helmet systems are estimating their respective 6-DOF pose accurately. Therefore, the deviations between these two scene points will serve as a good indicator of the performance of the overall AR-Weapon and AR-Helmet systems.

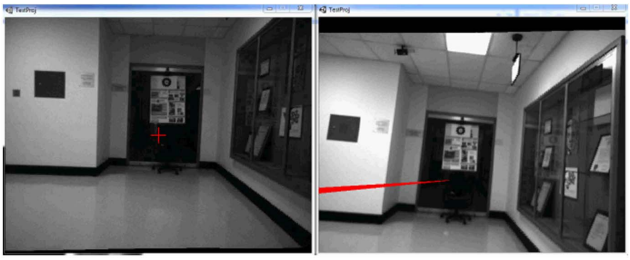

Figure 11: The live experiment setup.

Figure 12 illustrates how the scene point deviation is computed manually. Specially, we visually identified the scene point of the virtual weapon line is intersecting in the helmet image, and then marked it in the weapon image as P2 shown in Figure 12. Together with the cross center P1, the distance between P1 and P2 is computed as the scene point deviation. From the computed point deviation, the angular error can be computed directly via the pinhole camera projection model.

At each stop, we randomly selected a set of frames that aimed at the target for accuracy computation. From each frame, we first manually extracted the scene point deviation in pixels and then converted them into the angular errors.

Figure 12: The flowchart of computing the point deviation.

Figure 13 shows the computed average angular errors at each stop up to 55 meters, and the average angular error for all stops is around 6.56 mrads (approximately 0.3758 degrees). Under this angular error, a user will be able to hit a 36cm-diameter circular target by standing up to 55 meters away from it.

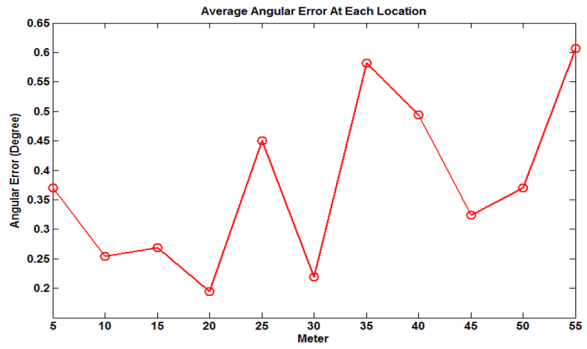

Figure 13: The computed average angular errors.

Figure 14 shows one sample of the selected frames at each stop up to 50 meters. In each image sample, the center of the red-cross in the weapon image represents the aiming target of the weapon, and the red line in the helmet image represents the estimated direction of the weapon. Visually, the estimated virtual weapon direction intersects with the target quite well.

7. Conclusions

In this paper, we have presented technical algorithms, system description and experiment results for a prototype AR-Weapon first-person shooting system. The system estimates 6-degrees of freedom orientation and location of the weapon and of the helmet of the user operating it, thus allowing the weapon to fire simulated projectiles for both direct fire and not line of sight during live runs. Using the estimated poses and a terrain model of the environment, virtual projectiles and synthetic virtual enemies are displayed on the user's head mounted OST display overlaid on top of the real world and full blast effects and simulated damages are displayed allowing the user to adjust fire accordingly. Experiment results demonstrated that our current AR-Weapon prototype system is able to achieve 6.56 mrads (approximately 0.3758 degrees) angular accuracy when shooting at the targets up to 55 meters. Future work will focus on further improving the angular accuracy and extending the range of the targets up to 100 meters during training.

ACKNOWLEDGEMENTS

The research reported in this document/presentation was performed in connection with the contract/instrument W911QX-13-C-0052 with the U.S. Army Research Laboratory through the University of Central Florida. The views and conclusions in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government or the University of Central Florida unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] H. Kato and M. Billinghurst. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. Int'l Workshop on AR, 1999.

[2] P. Muller. The Future Immersive Training Environment (FITE) JCTD: Improving Readiness through Innovation. Intraservice/Industry Training, Simulation & Education Conference (I/ITSEC), 2010.

[3] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar. Visual Odometry System Using Multiple Stereo Cameras and Inertial Measurement Unit. In IEEE CVPR, 2007.

[4] R. J. Fontana and S. J. Gunderson, Ultra-Wideband Precision Asset Location System. In IEEE Conference on Ultra Wideband Systems and Technologies, May 2002.

[5] G. Reitmayr and T. Drummond. Going Out: Robust Model-based Tracking for Outdoor Augmented Reality. ISMAR, 2006.

[6] D. Weng, D. Li, W. Xu, Y. Liu and Y. Wang. AR Shooter: An Augmented Reality Shooting Game System. IEEE ISMAR 2010.

[7] H. Cheng, R. Kumar, C. Basu, F. Han, S. Khan, H. Sawhney, C. Broaddus, C. Meng, A. Sufi, T. Germano, M. Kolsch, and J. Wachs. An Instrumentation and Computational Framework of Automated Behavior Analysis and Performance Evaluation for Infantry Training. In Proceedings of Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), 2009.

[8] Z. Zhu, H. Chiu, S. Ali, R. Hadsell, T. Oskiper, S. Samarasekera and R. Kumar. High-Precision Localization Using Visual Landmarks Fused with Range Data. In CVPR, 2011.

[9] T. Oskiper, H. Chiu, Z. Zhu, S. Samarasekera, R. Kumar. Stable Vision-Aided Navigation for Large-Area Augmented Reality. IEEE Virtual Reality, March 2011.

[10] M. Sizintsev, R. P. Wildes. Coarse-to-Fine Stereo Vision with Accurate 3D Boundaries. IVC, 28(3): 352-366, 2010.

[11] M. Sizintsev, S. Kuthirummal, S. Samarasekera, R. Kumar H. Sawhney and A. Chaudhry. GPU Accelerated Realtime Stereo for Augmented Reality. In 3DPVT, 2010
[12] B. Thomas, B. Close, J. Donoghue, J. Squires, P. D. Bondi and W. Piekarski. First Person Indoor/Outdoor Augmented Reality Application: ARQuake. Personal and Ubiquitous Computing. 6:75-86, 2002.

Figure 14: The sample of the selected frames for accuracy computation at each stop up to 50 meters.