Multi-Kinect Tracking for Dismounted Soldier Training

Brian M. Williamson and Dr. Joseph J. LaViola Jr.

University of Central Florida
Orlando, FL
bwilliam@eecs.ucf.edu, jjl@eecs.ucf.edu

Tim Roberts and Pat Garrity
U.S. Army Research Laboratory
Simulation and Training Technology Center
Orlando, FL
tim.e.roberts@us.army.mil
pat.garrity@us.army.mil

ABSTRACT

As low cost commercial video game sensors emerge, realistic full body interactions available in the household can also be utilized to support low cost dismounted Soldier training applications. These sensors, such as the Microsoft Kinect, are designed to work with users directly facing them. However, in an environment designed to train a team, larger spaces are necessary along with the freedom to maneuver and turn in all directions. These interactions are not reliably supported with the standard household video game configuration. In this paper, the use of multiple Kinects configured around a large area is examined, giving multiple Soldiers freedom of mobility and 360 degrees turning while wearing a Head Mounted Display. Skeletal recognition algorithms are shown within the Microsoft Kinect Software Development Kit that can be merged using commercially available tools and advanced fusion algorithms to produce better quality representations of users in the real world within a virtual environment. While one Kinect will often lose tracking of parts of a user, this paper shows that several Kinects coupled with inference algorithms can produce a much better tracked representation as users move around. Furthermore, the use of depth images along with the skeletal representations was examined to optimize fusion algorithms when bandwidth is available. Finally, it is shown how these techniques are capable of taking several skeletal representations in the virtual scene and merging them together to form a virtual representation of a single user. This system expands the viability of low cost commercial solutions to Soldier training in complex virtual environments.

KEYWORDS

Dismounted Soldier, Data Fusion, Virtual Environments, Full Body Interfaces, Immersive, Full Body Motions

ABOUT THE AUTHORS

Brian Williamson has a Master's in computer science from the University of Central Florida. His interests include computer graphics, video game design, user interfaces, military applications, and Global Positioning System / Inertial Navigation System navigation. His current project involves navigation techniques using the Wii Remote and MotionPlus, Microsoft Kinect, and Playstation Move technologies.

Joseph J. LaViola Jr. is an associate professor in the Department of Electrical Engineering and Computer Science and directs the Interactive Systems and User Experience Lab at the University of Central Florida. He is also an adjunct assistant research professor in the Computer Science Department at Brown University. His primary research interests include pen-based interactive computing, 3D spatial interfaces for video games, predictive motion tracking, multimodal interaction in virtual environments, and user interface evaluation. His work has appeared in journals such as ACM TOCHI, IEEE PAMI, Presence, and IEEE Computer Graphics & Applications, and he has presented research at conferences including ACM SIGGRAPH, ACM CHI, ACM Symposium on Interactive 3D Graphics, IEEE Virtual Reality, and Eurographics Virtual Environments. He has also co-authored "3D User Interfaces: Theory and Practice," the first comprehensive book on 3D user interfaces. In 2009, he won a National Science Foundation Career Award to conduct research on mathematical sketching. Joseph received a Sc.M. in Computer Science in 2000, a Sc.M. in Applied Mathematics in 2001, and a Ph.D. in Computer Science in 2005 from Brown University.

Timothy Roberts is a Science and Technology Manager at the U.S. Army Research Laboratory, Human Research and Engineering Directorate, Simulation and Training Technology Center. He currently works in the Dismounted Soldier Simulation Technologies conducting research and development in the area of dismounted Soldier training and simulation. His research interests include dismounted Soldier training environments, games for training, virtual locomotion, multi-modal interfaces, path planning, control systems and robotics. He received a B.S. in Electrical Engineering in 2002 and a M.S.E.E. in Control Systems and Robotics in 2007 from the University of Central Florida.

Pat Garrity is a Chief Engineer at the Army Research Laboratory, Human Research and Engineering Directorate, Simulation and Training Technology Center. He currently works in Dismounted Soldier Simulation Technologies conducting research and development in the area of dismounted Soldier training and simulation where he was the Army's Science and Technology Manager for the Embedded Training for Dismounted Soldiers program. His current interests include Human-In-The-Loop networked simulators, virtual and augmented reality, and immersive dismounted training applications. He earned his B.S. in Computer Engineering from the University of South Florida in 1985 and his M.S. in Simulation Systems from the University of Central Florida in 1994.

Multi-Kinect Tracking for Dismounted Soldier Training

Brian M. Williamson and Dr. Joseph J. LaViola Jr.

University of Central Florida
Orlando, FL
bwilliam@eecs.ucf.edu, jjl@eecs.ucf.edu

Tim Roberts and Pat Garrity
U.S. Army Research Laboratory
Simulation and Training Technology Center
Orlando, FL
tim.e.roberts@us.army.mil
pat.garrity@us.army.mil

INTRODUCTION

Low cost, non-obtrusive solutions for Soldier training have been shown feasible by adapting commercial products to complex problems (Williamson, et al., 2011). However, these off the shelf items are often built for simple applications. For example, in traditional video games, a user is always looking forward toward a television screen the tracking device is typically attached to. For dismounted Soldier training, however, Soldiers may wear a Head Mounted Display and view the simulation as they naturally turn within a training area.

With the Microsoft Kinect, the restriction to facing forward creates an unnatural user interface. It is a problem that can be resolved by adding more sensors in the area so the natural movements to control an avatar can be achieved so that the mantra "train as we fight" is ensured.

In this paper we discuss the RealEdge Fusion System, shown in Figure 1, which is a key piece to the next iteration of our RealEdge Prototype. The previous iteration showed the capability to solve locomotion and navigation problems with low cost solutions (Williamson, et al, 2011), but lacked an efficient 360 degree turning system. These techniques still remain well within the realm of low cost by only adding a few additional Kinect sensors and laptops to connect to them. The laptops used in the system perform some simple analysis of the Kinect data, then feed skeleton representations of the user to a centralized fusion server.

Two variations to fuse the data are examined. In the first approach, the orientation-less design does not require any gestures or extra equipment to determine the orientation of the user. It works under many scenarios, but the data is not at its highest fidelity for all scenarios of movements within the training area.

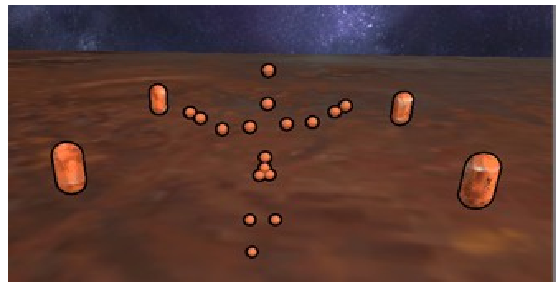

Figure 1: Example of Fusion System in the Unity scene. Each capsule represents a Kinect camera in the lab and the spheres represent skeleton joints.

In the second approach, it is assumed that the Soldier is holding a weapon that faces toward the direction they are facing. This results in better data fusion and orientation determination without adding any technology beyond the Kinects. However, the user is required to maintain their hands in front of their chest.

A test was designed for both approaches that showed the advantages of the fusion solution over the use of a single Kinect. As a user moved within a complete circle, our data shows the single Kinect losing tracking of key joints to the skeletal representation, while the fusion system remains steady in the expected data.

These fusion systems not only provide a Soldier full freedom of orientation within the training area's range, but also higher fidelity data as the multiple Kinects can be used to correct any failed data coming from any one individual sensor.

For this paper, four Kinect cameras are used. The Kinect, made by Microsoft, is a commercial video game device containing a Red-Green-Blue (RGB) camera and two 3D sensors. This creates two images, an RGB image, and a depth image. The Microsoft *Kinect for Windows* Software Development Kit (SDK) (Microsoft, 2012) was used, which takes the Kinect Data from the sensor and provides skeletal representations of all recognized users, RGB images, and depth images. In the RealEdge Fusion system skeletal representations were made use of as this was used in the previous RealEdge prototype for gesture recognition.

For the fusion system, the Unity 3D game engine (Unity, 2011), primarily used for video game development was selected as the virtual environment for the system. It provides a virtual environment that conveniently maps virtual unit space to meters. This was ideal for the RealEdge Fusion system as the Kinect SDK's skeletal representations are also in meter units. This allows the fusion system to place the joints of a Soldier within a 3D scene and align them in the virtual world. These virtual representations of the Kinect data can then be used for easy fusing, since all four representations are within the same coordinate system.

In the next section, work related to Soldier training and data fusion is discussed. Section three discusses the system built and the fusion iterations. Section four describes tests performed on the fusion systems, presents data from these tests and discusses the evaluation of the techniques. In section five the conclusions of this work and a discussion of future work is presented.

RELATED WORK

Virtual environments have been useful for Soldier training, for showing the capability to be trained to follow routes (Witmer, Bailey, and Knerr, 1995) and in later training iterations (Knerr, et al., 2003). However, the costs for high fidelity virtual systems can be very high, ranging up to hundreds of thousands of dollars (Knerr, 2006). As shown in the previous RealEdge prototype (Williamson, et al., 2011), solutions can be found that allow a Soldier training environment with non-obtrusive and low cost commercial solutions. We maintain these philosophies throughout this paper as well.

The concept of data fusion for military applications is not unique. As explained in (Hall and Llinas, 1997), data fusion has useful applications to the Department of Defense in the fields of ocean surveillance, air-to-air and surface-to-air defense, battlefield intelligence, target acquisition, and strategic warning and defense. Typically, the need for data fusion is for multiple sensors with multiple specialties to combine their data into an image of the scene better than their individual pieces; a system with a natural analogy to how the human brain combines information from the multiple senses (Varshney, 1997). The process model for multi-sensor data fusion includes pre-processing, data alignment and correlation, object aggregation, force estimation, and performance evaluation (Hall and Llinas, 1997). In the RealEdge Fusion system, since the same type of sensor is used, several issues to data fusion do not apply. Instead, the focus was on data alignment and object correlation.

Ever since the Nintendo Wii Remote's application programming interface became available (Peek, 2008), it has shown the capability of commercial devices to be used for virtual environments (Lee, 2008) (Shirai, Geslin and Richir, 2007) (Wingrave, et al., 2010). This has resulted in the creation of other 3D user interface devices, including the Playstation Move (Sony, 2011) and the Microsoft Kinect (Microsoft, 2012). In the past few years, the Kinect has been shown to be vital to various areas of research, from real time traffic sign recognition (Par and Tosun, 2012) to the exploration of medical data (Gallo, Platicelli and Ciampi, 2011).

Furthermore, the concept of using multiple Kinects to fuse data has emerged in the research community, especially in the Kinect Fusion system (Izadi, et al., 2011). Their focus was on refining the depth images of multiple Kinects to reconstruct an entire scene in a virtual environment and interact with the scene. In this paper, the focus relied on fusing the actual skeletal representations rather than the depth images, as this representation has been shown useful to gesture recognition. There is also research on the use of multiples Kinects having depth images combined and used with Hidden Markov Models to track dynamic objects within a static scene (Dubois, Dib and Charpillet, 2011).

Previous research shows that the fusion of multiple sensors to create a single understanding of a scene has been performed for years with success. Furthermore, fusion of depth image data with the Kinect has been emerging into the research community. It is with this knowledge that the RealEdge Fusion system was created, capable of fusing skeletal representations from four Kinects into a single understanding of the scene.

REALEDGE Fusion

System Setup

The fusion system was composed of three Dell Inspirion N7110 laptop computers and four Kinect cameras attached to generic camera tripods. The tripods were chosen as they provided stability, adjustability to various Soldier's heights, and angle markings to know precisely the direction of the sensor. A Dell Precision T3500 desktop was used as the fusion system and the connecting point for one of the Kinect cameras. Multiple computers had to be used as, at the time the fusion system was started, the Kinect SDK did not allow multiple Kinects to perform skeletal recognition on the same PC.

Figure 2: Example of a Kinect attached to a camera tripod.

The streaming software was run on each laptop and the fusion system. The software would connect to the Kinect sensor via the SDK, and as skeleton and depth image data was received it would perform basic analysis on the data before streaming a custom structure to the fusion system via Transmission Control Protocol, as seen in Figure 3. The custom structure was a refactoring of the skeleton representation along with pre-analyzed data. There was also the capability to request a depth image frame to display, though this was used only for debugging.

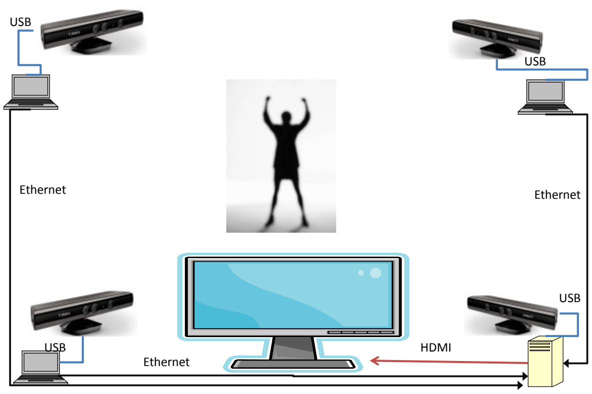

Figure 3: Architectural diagram of lab setup showing three Kinects going to laptops and one going to the central server, all transmitting via ethernet.

Pre-Distribution Data Analysis

Before the data was transmitted to the fusion server, the streaming system was tasked with analyzing the skeleton and depth data from the SDK in order to determine some level of confidence in the joint systems. The SDK provides means to transform skeletal coordinates back to the depth image, and to transform depth pixels into the skeletal coordinate system.

In a first iteration, it was found that the depth image could be sliced to contain just the pixels containing a user, by performing coordinate transformations of joints to pixels and subtracting the background (pixels far away). Each pixel could then be taken and translated to the skeletal coordinate system to find the closest skeletal joint near it. This allowed the depth image to be segmented to each joint. Ultimately, this process limited the laptop's ability to process frames at the Kinect's 30 frames-per-second, so it was excluded from the final RealEdge Fusion system. It did, however, show future potential for analyzing each joint's depth pixels to verify the skeletal representation.

In the end, a joint confidence system was created off of whether the joint was classified as tracked (1.0), inferred (0.5), or not-tracked (0.0) by the SDK. An analysis was then performed where the skeleton joint was transformed to the depth image coordinates, and then the image's pixel was translated back to skeletal coordinates. If the distance between the original skeleton joint provided by the system and the one provided via translating from the depth pixel was large, adjustments would be made to the confidence in that joint slightly. Because more analysis is necessary for this technique and cause for this variation, the adjustments to the confidence were not largely based off of it; just enough to suggest an irregularity was seen.

Once the confidence for each joint was determined, the streaming software would then create a custom structure filled with the skeletal representation and confidence values. This structure was then serialized and streamed to the fusion system.

In the Fusion system, Unity 3D was used to recreate the physical lab setup in the virtual world. Four Kinect representations were placed and oriented in relation to the real world cameras (measurements taken in meters). Then, as the data was received, the Kinect representations would create skeletal structures made of spheres naturally projected at the offsets given originally by the Kinect SDK. Since the orientations and measurements matched real world configurations, the four skeletal structures in Unity overlapped each other, as seen in Figure 4. Though not included in this paper, potentially multiple users could be recognized by this system, as skeletal representations near each other very likely represent the same person.

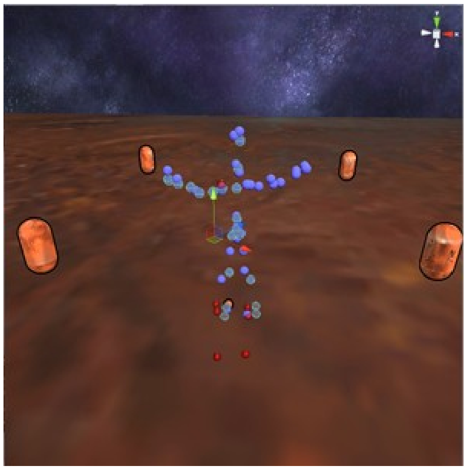

Figure 4: The 3D Unity scene with each Kinect's skeletal representation shown as spheres in blue, which are overlapping naturally within the scene.

Orientation-less Data Fusion

The first goal was to create a fusion system that worked for all scenarios and did not require the user to have any special equipment or gestures to configure. This was named the orientation-less data fusion approach and went through three iterations.

The first iteration of the orientation-less system was used primarily to illustrate the challenges that would be encountered. The orientation-less system simply took the joint data from every Kinect, and if over a certain confidence level (0.75), it would perform a weighted average of the joint's position as shown in

$$\hat{X}_j = \frac{\sum_i W_{i,j} * X_{i,j}}{\sum_i W_{i,j}} \quad (1)$$

where $\hat{X}_j$ is the averaged position of the joint j, $W_{i,j}$ is the weight of joint j from Kinect i and $X_{i,j}$ is the position of joint j from Kinect i.

This iteration showed an immediate issue between the representation of left and right between the sensors. The Kinect SDK assumes the user is facing the sensor, so the Kinects behind the user would swap left and right joints. This means when computing the position of the right hand joint in the virtual representation, the position would actually be of the left hand in the data.

To resolve this came the second iteration, which did not combine the joints in (1) based on their representation in the skeletal structure, but instead looked at the position of the spheres representations in Unity. Since the skeletal representations were very near each other, iterations through each joint from the first Kinect sensor was used as the basis, then all spheres in the Unity scene near the reference one were found and used to perfect the tracking of that joint.

This worked in practice, except if joints came too near each other. For example, if the user crossed their arms or put their hands on their hips, this would cause the joints to get confused with the torso joint or hip joints and shift erratically.

The third iteration attempted to resolve the left-right issue by just swapping all of the joints for two of the Kinects, considered behind the user at startup. Though while facing the Kinects, their data would still be swapped, this was determined to be fine as the originally front-facing Kinects would now be swapping as they are looking at the back of the user. The goal was to have consistency between the Kinects.

A voting system was then implemented, which went through each joint and looked for the maximum number of Kinects to agree on a position. Then the weighted average between these close positions was used. The idea was that it was commonly seen that two to three Kinects were in agreement, with the other two being vastly apart or admitting they were inferring the joint's location. By looking for this agreement adjustments could be made using that joint's location.

This iteration gave a reliable skeleton capable of gesture recognition. However, joints would still shift in certain locations of the training area, caused primarily by the wrong two Kinects agreeing over the correct one. While these issues were rare, it was finally concluded to attempt an orientation based design.

Orientation Based Data Fusion

In this approach, the context of a Soldier in training was used and it was assumed that a weapon would be in their hands aiming in the direction they desire to face. While not true for all scenarios, it did present an opportunity to determine the orientation of the user and perfect the fusion system.

The hands were examined of each skeletal representation, looking for the ones most forward from the torso as determined in

$$D_h = \sqrt{\left(J_{h,x} - J_{c,x}\right)^2 + \left(J_{h,z} - J_{c,z}\right)^2} \quad (2)$$

where D_h is the distance from the hand h and the torso c considering only the x,z plane, $J_{h,x}$ is the x position of the hand joint, $J_{h,z}$ is the z position of the hand joint and $J_{c,x}$ and $J_{c,z}$ are the similar positions of the torso joint.

Using only hands farther away from the chest, in the event that one has returned to the belt or to perform some other gesture, the orientation could then be determined from the hand to chest with the arctangent function. Furthermore, the hand joint's confidence had to be over a set amount (0.75) to be considered. With an orientation reported from the arctangent function, the angle was remapped to the [0,360] degree range. If the Kinect could not determine orientation because of low confidence in hands, a -1 value was returned and checked for.

With each Kinect calculating an orientation value, a simple voting method was used to find two or more Kinects agreeing on an orientation and used it. Initial tests showed this data to be highly accurate to mapping to where the weapon was pointing. Each Kinect was then hardcoded to the angular range it is best suited to detect based on its location within the lab. With these two pieces of data, the system was able to detect the two best Kinects for detecting the user, to which it then used a weighted average formula similar to (1) to determine the position of each joint of the user.

This system showed high fidelity and range of movement through the entire set of tests. Satisfied with the high fidelity of this fused data, it was decided to analyze the systems in comparison with a single Kinect.

EVALUATION

To evaluate the techniques of this system, the position of a user's arms were tracked as they moved within a complete circle standing at the center of the recognition area. For all data collection the arms were held up in a straight line, perpendicular to the main axis of the body. This allowed the Y position of each arm joint to stay relatively stable as the user made the full circle. The X and Z plane would change in accordance to making a circle.

All data recorded was based on the virtual scene's coordinate system so that there was consistency among all of the Kinects. Each Kinect and fusion system's data was recorded for both orientation-less and orientation based systems. The voting method was used for the orientation-less system.

Shown in Figure 5 and Figure 6 are examples of the two fusion approaches for a particular joint's Y value as the user turned in a full circle. The fusion approach line has been enlarged for illustration. Most Kinects by themselves are able to keep a steady track of the joints, except for when view of the user is lost. When this occurs, drastic jumps in the data can be seen. Both fusion approaches show a steady line, able to interpret through any sudden jumps in the data.

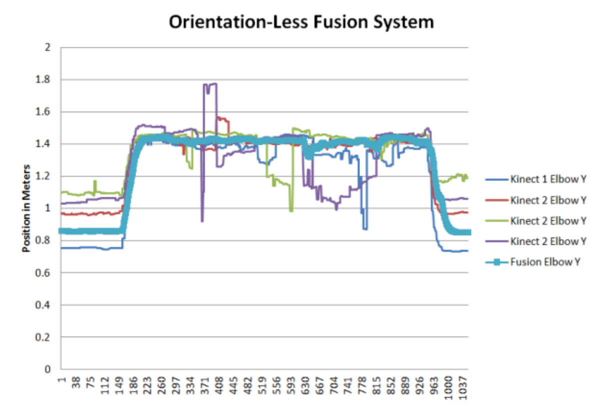

Figure 5: Orientation-Less Fusion System Example

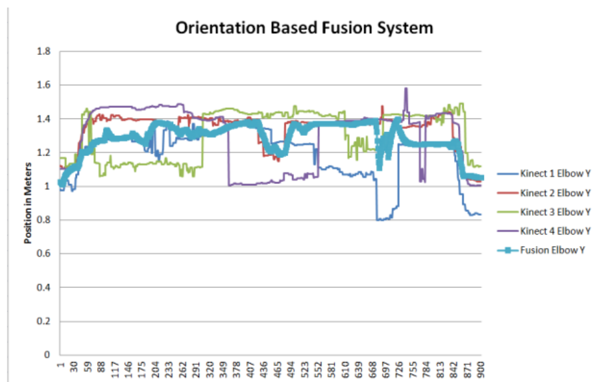

Figure 6: Orientation Based Fusion System Example

The relative error for each system was also evaluated by using the average height recorded for the elbow, wrist, and hand joints as an estimation to the measurement. In Tables 1 and 2, the maximum errors present are shown, which demonstrate the single Kinect systems jumping largely as the joints are lost in sight and no longer inferred, while the Fusion system shows a significantly smaller amount of error. In some scenarios, as seen in Table 2, Kinect 2, the inference algorithms of the Kinect can come through and estimate the joint while not maintaining visibility. In these tables the numbers 5, 6, and 7 represent the elbow, write and hand joints of one arm with 9, 10, and 11 being the same joints for the other arm. All of these calculations come from the Y position of the joint.

Table 1: Orientation-Less Fusion System Max Relative Error

	5.y	6.y	7.y	9.y	10.y	11.y
K1	0.362	0.1720	0.199	0.4003	0.2973	0.3070
K2	0.1054	0.0736	0.0761	0.0736	0.1219	0.1486
K3	0.3073	0.2922	0.2368	0.2453	0.512	0.6006
K4	0.3254	0.4551	0.4661	0.4584	0.4741	0.4562
Fusion	0.0683	0.0532	0.0722	0.0973	0.1025	0.1318

Table 2: Orientation Based Fusion System Max Relative Error

	5.y	6.y	7.y	9.y	10.y	11.y
K1	0.3462	0.4151	0.3992	0.3744	0.2349	0.1661
K2	0.1618	0.0698	0.0763	0.1935	0.1187	0.1131
K3	0.1912	0.4199	0.4365	0.192	0.4668	0.5604
K4	0.2358	0.4406	0.533	0.0965	0.0969	0.0842
Fusion	0.1596	0.0929	0.1338	0.1933	0.0984	0.0977

While the orientation based and orientation-less techniques look very similar in these tests, in practice with a user performing quick movements to random spots, the orientation based approach is visually seen to track them better than the orientation-less solution. This data does show, however, that even without knowing the orientation of the user, data can still be fused successfully for several scenarios.

CONCLUSION AND FUTURE WORK

A solution for fusing skeletal representation data from multiple Kinects to provide complete coverage of a user was successfully demonstrated. Where a single Kinect would fail to see a user at certain angles, both of the systems discussed in this paper show capability of being able to do so. The fused skeletal representation is capable of being used in simulation training for gesture recognition and navigation needs, as though it came from an original single Kinect.

There is still work to be done to perfect such a system. As mentioned previously, computational requirements prevented analyzing the depth pixels for every joint in order to determine high fidelity confidence in the Kinect's representation. In future work, the 30 frames per second constraint can be placed aside to perform analysis and see what data is present, and then optimizations to performance can be reintroduced.

The possibility to track multiple users by using the proximity of their skeletal systems in the virtual scene was also stated; however, this work was not discussed in this paper. In the future it will be shown that this can be done simply and easily with the Unity system. Finally, the fused skeletal representation needs to be placed back into a training environment so that gesture recognition can be utilized in a 360 degree environment.

While there remains much work, these techniques present a viable low cost solution for a commercial product constraint of only working when the Soldier is facing forward. The researchers were able to continue using the same product and perform data fusion in order to create a reliable single representation of the Soldier. Furthermore, capability was shown to track the Soldier's orientation with no additional equipment than the weapon they would already be holding during training.

REFERENCES

Dubois, A., Dib, A., Charpillet, F. (2011). *Using HMMs for Discriminating Mobile from Static Objects in 3D Occupancy Grid*. 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pages 170-176

Gallo L., Placitelli, A.P., Ciampi, M. (2011). *Controller-Free Exploration of Medical Image Data: Experiencing the Kinect*. 24th International Symposium of Computer Based Medical Systems (CBMS)

Hall, D., Llinas, J. (1997). *An Introduction to Multisensor Data Fusion*. Proceedings of the IEEE, Vol. 85, Issue 1, pages (6-23)

Izardi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A. (2011). *KinectFusion: Real-time 3D Reconstruction and Interaction using a Moving Depth Camera*. ACM Symposium on User Interface Software and Technology, October 2011

Knerr, B., Lampton, D., Thomas M., Comer, B., Grosse, J., Centric, J., et al. (2003). *Virtual Environments for Dismounted Soldier Simulation, Training, and Mission Rehearsal: Result of the FY 2002 Culminating Event*. U.S. Army Research Institute for the Behavioral and Social Sciences

Knerr, B (2006). *Current Issues in the Use of Virtual Simulations for Dismounted Soldier Training.* U.S. Army Research Institute for the Behavioral and Social Sciences

Lee, J. (2008). Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing* , pages 39-45

Microsoft (2012). *Developer for Kinect*. Retrieved June 10th, 1012 from http://www.microsoft.com/en-us/kinectforwindows/

Par, K., Tosun, O. (2012). *Real-Time Traffic Sign Recognition with Map Fusion on Multicore/Many-core Architectures*. To appear in Acta Polytechnica Hungaria, Volume 9, November 2012

Peek, B. (2008). *WiimoteLib -.Net Managed Library for Nintendo Wii Remote*. Retrieved September 20, 2009 from http://www.brianpeek.com

Shirai, A., Geslin, E., &Richir, S. (2007). WiiMedia: motion analysis methods and applications using a consumer video game controller. *SIGGRAPH Sandbox*.

Sony (2011). *Move.Me | Playstation Move*. Retrieved May 15th, 2011 from http://us.playstation.com/ps3/playstation-move/move-me/

Unity (2011). *UNITY: Unity 3 Engine Features*. Retrieved May 3rd, 2011 from http://unity3d.com/unity/engine/

Varshney, P.K. (1997). *Multisensor Data Fusion*. Electronic and Communication Engineering Journal, Vol. 9, Issue 6, pages 245-253

Williamson, B., Wingrave, C., LaViola, J.J., Roberts, T., Garrity, P. (2011). *Natural Full Body Interaction for Navigation in Dismounted Soldier Training.* Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)

Wingrave, C., Williamson, B., Varcholik, P., Rose, J., Miller, A., Charbonneau, E., et al. (2010). Wii Remote and Beyond: Using Spatially Convenient Devices for 3DUIs.*IEEE Computer Graphics and Applications*, pages (71-85).

Witmer, B., Bailey, J., Knerr, B. (1995). *Training Dismounted Soldiers in Virtual Environments: Route Learning and Transfer*. U.S. Army Research Institute for the Behavioral and Social Sciences