



Real-Time Motion Capture on a Budget

Tami Griffith^{1(✉)}, Tabitha Dwyer², and Jennie Ablanedo³

¹ Army Research Laboratory Simulation & Training Technology Center,
Orlando, USA

tamara.s.griffith.civ@mail.mil

² Cole Engineering Services, Inc., Orlando, USA
tabitha.dwyer@cesicorp.com

³ Institute for Simulation and Training, University of Central Florida,
Orlando, USA

jablaned@ist.ucf.edu

Abstract. The U.S. Army Research Laboratory's Simulation & Training Technology Center, along with Cole Engineering Services, Inc. and the University of Central Florida have set out to leverage commercial technology with the goal of improving realism, and reducing cost for Army training tasks. The focus of this task is to establish a prototype functionality that allows a live person to take control of a virtual character. This is done using the Enhanced Dynamic Geo-Social Environment, which is an Army-owned simulation built upon the Unreal Engine 4.

Commercial games and movies make use of motion capture capabilities to animate characters. This functionality is needed in real-time to allow person-to-person interactions within a simulation. The goal is to have puppeteers that can take over Artificial Intelligence (AI) characters when in-depth interactions need to occur. While AAA games and movie budgets allow for more expensive systems, the goal of this team is to keep the cost well below \$10,000.

A market analysis along with this team's experience utilizing and integrating the market capabilities to meet these goals are described in this paper.

Keywords: Virtual training · Avatar puppeteering · Virtual humans
Real-time motion capture

1 Introduction

1.1 Background

Motion capture systems are frequently used in the entertainment industry to create an elegant and fluid animation of characters for both movies and games. These systems typically create an accurate and natural representation of a body's dynamic motion. The captured animations are later hand-polished by teams of artists, creating the seamless and natural movement you commonly view within animated films and games. The authors are interested in making use of commercial motion capture capability, including both facial expressions and body movement, to support real-time training experiences. Despite the ubiquitous nature of motion capture systems, it is still unusual for these systems to be used in real-time without an artist to polish the resulting

animations. Because of this unique use case, the hardware and software options have been very limited. A small number of commercial companies at the Game Developers Conference in 2015 promoted real-time motion capture. One of those companies offered a solution that cost roughly \$30,000 a year, while another was \$1500. The latter was the solution the team pursued for the first prototype of full body and face tracking in real-time. The costs of systems and the equipment needed and can be very expensive to purchase. The authors hoped to make use of low-cost options, though some of most effective real-time body tracking solutions and hardware (such as professionally engineered helmets and cameras) come at a relatively high cost.

This paper describes the research from both a retrospective and future outlook on creating a cost-effective real-time motion capture system, or real-time puppeteering system. Our focus includes integrating real-time motion capture software and hardware into a training game utilizing the Unreal Engine 4.

This paper will also describe analysis of both positive and negative outcomes in creating an overall system. It provides justification for hardware and software solutions and documents how the system evolved through several prototypes. A thorough investigation on affordable real-time body and facial tracking products is described. The initial attempt solely focused on face tracking, without the avatar's body, using a commercial program that is no longer available. The solution evolved to artist-created body gestures/animations, controlled by an Xbox 360 controller. This solution proved unwieldy and felt unnatural for the actor controlling the puppet. Movements were also limited. This resulted in the team exploring ways to improve the prototype. For example, the team experimented with different headgear to keep the camera in front of the actor's face at the appropriate distance and with the right amount of light. A bike helmet and tactical helmet were tried, but there were issues with fitting various users and keeping eyebrows visible to the camera. Most recently, adjustable headgear used for wrestling was used to great effect. This headgear is reasonably priced, and has provided clear view of facial features as well as eyebrows and can easily be adjustable to fit nearly all sized heads. One member of the team modeled hardware and 3D printed it to function as camera mounts on the helmet, and we've used a range of cameras as simple as basic webcams increasing in complexity and cost to \$300 depth cameras.

The current prototype uses the HTC Vive with inverse kinematics to resolve body tracking in real-time. We are using a popular, and somewhat affordable, commercial product for face tracking. We also have a working relationship with another facing tracking company that is building a plug-in to support this application in Unreal Engine 4.

Finally, this paper includes a forecast on other solutions that are emerging on the horizon. In particular, Ninja Theory, Ltd is providing good documentation on their motion capture strategy with the game *Hellblade: Senua's Sacrifice*. The developers of this game attempted to use low-cost capture products, but ultimately ended up using high-end products.

The technology is still in its infancy, and the results have not yet matured into a seamless experience for training. The details of planned future work is included in this paper. This paper is being developed as part of a strategy to grow a community-of-practice into the use of real-time tracking.

1.2 Real-Time Puppeting

The realism of interpersonal interactions in virtual training domains can be an important factor that influences the decisions of a learning population. Some training tasks require very detailed realism, including small-motor movements, eye movements, voice intonation and the ability for avatars to gesticulate while walking in the virtual space. The interaction between automated virtual agents does not yet approach human-to-human experiences. This drives our team’s interest in increasing the realism of human interactions in virtual environments by using real-time puppeteering [1].

This capability leverages investments across multiple government agencies to optimize costs. The focus is on merging state-of-the-art commercial technologies to support human-dimension training for force effectiveness. The working prototype provides a platform for feedback from end-users and informs the requirements and procurement communities. The authors would like to demonstrate whether this capability is possible at a very low, off-the-shelf cost, with no limits to the number of end-users and no per-seat end-user fees.

There are many potential applications for this technology. For example, Human Resource professionals could use virtual interactions to hone skills in recognizing signs of Post-Traumatic Stress Disorder (PTSD), or indications an individual has experienced sexual harassment or sexual assault. While human-to-human interaction is “the gold standard” to train these skills, the virtual component allows a small number of actors to play various roles to explore racial, gender, sexual orientation, or age discrimination issues. This capability can also be used to breathe life into a virtual landscape; providing the sense that a town is teeming with life. Artificial Intelligence (AI) could control patterns of life within a virtual town. Then as trainees encounter a character, an actor could step into that role and support natural conversation. An entire town could be managed with only a small set of human puppeteers since multiple avatars can be controlled by one person.

2 Stakeholder Goals

2.1 Army Research Laboratory Simulation and Training Technology Center (ARL STTC)

The U.S. Army Research Laboratory Simulation & Training Technology Center (ARL STTC) is focused on research on strategies to improve training for U.S. Army Soldiers. ARL STTC has been exploring using commercial game technology as a way to reduce development time in building simulations to support training tasks. The Enhanced Dynamic Geo-Social Environment (EDGE) is one such simulation platform. EDGE is currently built on the Unreal Engine 4, though it has made use of the Unreal Engine 3 platform in the past. EDGE has been shared across multiple Government agencies with investments from the Department of Homeland Security, the Defense Equal Opportunities Management Institute and the Federal Law Enforcement Training Center, to name a few. The platform is used to explore ways to take emerging technology and apply it to various training tasks. The Army makes use of Computer Generated Forces (CGF) or Semi-Automated Forces (SAF) entities to function as the opposing forces in

traditional simulations. Making use of game engines, we are able to make use of Artificial Intelligence (AI). While the AI that comes as part of a commercial game is able to make rudimentary decisions based on the terrain and respond to simple scripted dialog, it is quickly evident that the characters are not driven by a live person. Natural language interpretation is better, but nuances in language still limit verbal interchanges. Until AI is good enough to be used for human-to-human interactions in a simulation, puppeteering can be used to meet the need.

2.2 Impact to the Army

The Army currently supports a great amount of live training events. In some cases, actors are paid to play the role of people in a town. They may even have farm animals such as goats and chickens wandering the town to provide a sense of realism. Soldiers can watch the patterns of life within the town to look for anomalies. Then they can enter the town and speak with passersby, a key leader or law enforcement. As you can imagine, these live events can be costly and logistically taxing. Imagine the cost savings if the town could be living and breathing in a persistent state of a virtual environment when the soldiers log in. A couple of role-players, maybe one female and one male, could hop into a character as a Soldier approaches them, but could be controlled by AI otherwise. Cost savings are important, but it cannot be at the expense of training capability. Our intent, with this research, is to make the virtual environment so that it provides the same training capability as is possible in the live environment to reduce overall training costs to the Army. This represents just one use-case of how this technology can benefit Army training.

3 Prototyping

3.1 Research Problem

The research in real-time motion capture, or puppeteering, initially began to bridge the gap between conversations with a real human and conversations with artificial intelligence (AI) [1]. While great strides are being made in having a natural conversation with an AI character, [2] there is still a large gap in natural response and emotion [3]. Thus, our group began research into real-time motion capture of a human actor puppeting a virtual character, providing the virtual character the natural motion and emotion needed for conversation while at the same time allowing the actor to portray many different characters in a game environment [1].

Budget constraints inspired the goal to reduce end-user costs to achieve a similar real-time motion capture result. Can an effective Real-Time Motion capture solution be achieved on a limited budget? What levels of success can be repurposed for other Government user's? This paper will discuss our path toward finding answers.

3.2 Commercial Game Studio Study

In 2014, in the rudimentary phase of the puppeteering research, the commercial company, Ninja Theory, Ltd, was making similar budget considerations on their new game, Hellblade: Senua's Sacrifice [4]. Ninja Theory, famous for making such games as Heavenly Sword, and DmC: Devil May Cry, decided the company wanted to self-fund their newest game, Hellblade [5]. To successfully accomplish the self-funding goal, Ninja Theory sacrificed several budgetary trade-offs during development [6]. While their proposed budget of just under \$10 million is still orders of magnitude larger than our budget, it was beneficial to see the similarity in choices both teams were making [7].

Ninja Theory's decisions in relation to motion capture showed promise for development for the parallel of puppeteering research. Many of the choices made for the puppeteering, including shopping for lights on Amazon, 3D printing camera hardware, using GoPro cameras and mounting hardware, were similar decisions [8]. Ultimately, through trial and error, the puppeteering research, saw some of the same failures, or unacceptable results, that they found with their low budget solutions [1]. Ninja theory, ultimately, borrowed expensive hardware and software from Vicon Motion Capture Products to complete their project [9]. This included high-end motion capture cameras and a professional motion capture helmet, which alone can cost over \$3000 [9]. There were many lessons learned through the shared experiences as Ninja Theory explored current technology to meet their goals.

3.3 Software

There were a variety of software components that came together to provide the desired functionality. Code is needed to link the face and body tracking systems to the game engine. The game engine provides the foundational environment and development tools. EDGE is Government owned software built on the UE4 Game engine that allows the researchers to experiment with various technical solutions [10].

Game Engine. The foundation of development is based on the Unreal Engine 4 (UE4) game engine, which many AAA games have been built. The Enhanced Dynamic Geo-Social Environment (EDGE) was built on UE4 and was used for a variety of prototypes. The puppeteering effort is one such prototype, stemming from the concept of filling a gap between human-to-human and human to Artificial Intelligence (AI) conversation [1]. The Unreal Engine 4 was chosen for development for a several reasons:

- The team had significant experience in development in the engine, dating back to prototypes this team developed on Unreal Engine 3. We have been developing in UE4 since 2013 when it was still in beta [1].
- Since our games and research do not generate profit, we are able to use the engine at no cost. One reason we use this game engine is the full access to all source code. The development community is quite large and active, through which we have found many examples and answers to questions through the Unreal Answers

website and the community forums. There is also a marketplace where we have bought 3D model assets and gameplay features providing cost savings over having to develop ourselves.

- The platform UE4 is also a common platform supported by third party commercial developers. An example of this would be Speed Tree, which builds and ages foliage, plugging directly into the game engine. This is important, because many of the companies currently contain an already developed UE4 plugins or code examples ready to integrate. This is extra work that may be required if we were using a different game engine.

Enhanced Dynamic Geo-Social Environment (EDGE). Virtual training simulations can provide cost efficiencies while also improving training outcomes when used in a blended training concept [10]. Virtual training cannot replace the interaction involved in live training; however, there are opportunities to significantly reduce costs while increasing responder proficiency by applying technologies to support training strategies.

EDGE is a virtual platform developed by the U.S. Army Research Laboratory’s Simulation & Training Technology Center (STTC) in partnership with the Training and Doctrine Command (TRADOC), the Department of Homeland Security (DHS) and various other agencies. EDGE is a government owned prototype designed to provide a highly-accurate virtual environment representing the operational environment and utilizing the latest gaming technologies. This collaborative government prototype leverages investment across multiple government efforts to maximize efficiencies (cost savings) by exploiting emerging technology [11].

3.4 Developmental Stages

The puppeteering work has been through many stages of development for over a year and a half. Initially, the goal was simply to accomplish human facial tracking in real-time inside of UE4. Development progressed to adding body animation, and eventually full body tracking. The following section details the stages of development up to the writing of this paper. Earlier work in this area is described in previous papers [1].

The team wanted to answer the question, “Can we convey human emotion in conversation through a virtual avatar?” We thought we could accomplish this through facial tracking software and a camera. We had used software in the past for tradition motion capture and knew it had a real-time component. The team looked toward the commercial market and assessed existing technology. Low-cost options in this space are rare, since they generally target big budget game and movie studios.

Test #1: Facial Capture Only. This effort started with a tracking program that used the Xbox Kinect hardware. The program was developed by a single person and was quite inexpensive. The Xbox Kinect is a depth camera, and we believed that would provide higher fidelity tracking. The integration with UE4 turned out to be labor intensive, and we never made it past the initial integration and test because the tracking software didn’t track eye movement. Losing this key feature (eye tracking) removed the sense of realism from the avatar since the eyes were always locked straight ahead. From this task, we learned the importance of tracking eyes.

Test #2: Facial Capture Only, Different Capture Software. The only off-the-shelf options remaining were significantly more expensive, however most companies offered a brief free trial period. Additionally, many companies offered educational or independent video game or Indie (i.e. a game developed without the financial support of a publisher) [12] discount. This allowed for initial testing, and comparison of each of the necessary products. Since the research and prototyping is not for profit, this helped tremendously in accomplishing market research.

The next choice for facial tracking was a larger commercial company discovered at the Game Developer Conference. Their software product also made use of depth cameras for the tracking hardware. The software was tested with the Xbox Kinect. Simultaneously market research was conducted on other depth cameras. There were limited depth cameras on the market, and they were almost five times as expensive as a standard web cam.

While each solution required software development, artwork changes, and hardware creation and/or setup, this capture software required our game characters to have 51 morph target, or shapes, [13] built into the face to control. The art team took existing characters, modified the skeletal mesh to incorporate the shapes, and reimported the character back into UE4. The tasks were coded to an Application Programming Interface (API). The software provided a network stream of shape values for each render frame, and our development team added code to apply those shape values to the character's face in real-time. The camera was placed on a tripod in front of the computer monitor, with the actor sitting and facing the camera.

The initial test with this face tracking solution, as stated, used the Xbox Kinect, but the results were less than ideal. The focal length of the Kinect is very far [14]. This works well for tracking a body, but when focusing on the face, the actor had to sit far away from the desk and camera. This made it challenging to pick up the nuanced movement of the face and eyes. Not knowing if this was a software tracking problem or a hardware (camera) problem, we decided to purchase the depth-camera recommended by the vendor for further testing.

Using the new depth camera, results improved significantly. Exchanging the camera was simple using a dropdown list of cameras within the tracking software. This tracking software required an in-depth calibration for each actor, having them go through a series of expressions and saving that calibration data into a file. It soon became clear that this step was extremely important in future work. After calibration and running with UE4, the results of the expressions and emotion conveyed well through the game avatars.

Test #3: The Desire for Body Language. While face tracking was successful, it was very limiting in scope. For example, the camera was focused on the shoulders and head while the lower body was shown with an idle animation. If the actor expressed anger or sadness, the emotion was betrayed by the body not moving in a sympathetic way. That led to the desire to add body language in concert with the facial tracking.

The original concept did not involve full body real-time motion capture. Rather, initially, the concept was that the actor would function using a game controller to control the body. This mimics other projects and puppeteering throughout history,

much like the way Jabba the Hutt's facial features were controlled by radio control in *Star Wars: Return of the Jedi* [15].

The development effort for this concept was fast paced. The art team created a set of polished character animations for several selected emotions, or postures. The development team mapped those animations to a set of controls on a game controller, and created a user interface that reflected the controlled animations. The design philosophy was to use simple emoticons on-screen and have them mapped to combinations of button presses. For instance, to have the virtual avatar appear nervous, the actor would hold the right trigger and press down on the directional pad of the controller.

The goal was to have this method of acting be intuitive for the actor, while allowing for clean and believable animations. In practice, there were clear benefits and drawbacks to this strategy. Using this strategy, the body gestures could be made to match the facial expressions. However, there was only a finite number of different expressions. The set is nowhere near limitless as one would see in real-life. The biggest issue, though, was that no matter how intuitive the controls were, it was not a natural behavior for the actor. They had to remember that if they were going to say something and appear nervous, they had to press a button combination to get the body into that state. This made it hard to improvise. The actor might respond to a question and quickly look and sound angry on their face, but might forget to change the body using the controller, thus breaking the sense of immersion.

The animations, however, were very clean, having benefited from animator post-processing, unlike raw motion capture data. It was possible to make gestures such as the hands covering the face, which you couldn't do if the actor was covering their face from the camera due to the arms not being collision bodies. This is an issue the team still struggles with today. If arms do not collide they tend to move through one another breaking the sense of realism.

Test #4: Combining Real-time Body with Real-Time Face. Research and development evolved to strive further for a more natural method to capture the body without the controller. As a result, it was decided to find a similar solution of a real-time motion capture system for the body comparable to the one used for the face. To date, the team is unaware of any one solution that handles both, so the focus moved to compile the real-time body capture with the real-time facial capture. Although there has been basic research using the Kinect, as a real-time controller, the research was rudimentary [16], and unfortunately, there were no apparent solutions that would easily integrate into UE4 without significant time and money that would need to be invested.

At the Game Developer Conference additional companies showcased exploring their solutions to the technical field of Real Time Motion Capture. Commercial game companies were now toying with the idea that they could use real-time motion capture to preview their character's performances in the actual game environment [2]. This capability has the potential to save considerable time (and by association money) while shooting motion capture activities [1]. The director can assess the quality of the capture before investing time cleaning up the animation. This capability motivated motion capture hardware and software developers to offer more options for supporting real-time previews [9].

There are two significant drawbacks to most of the vendors providing real-time body motion capture: footprint and price. For many companies, providing a real-time capability does not require additional hardware. A tradition motion capture studio setup involves at least 20 square feet of space, and includes scaffolding and an expensive array of cameras. Our use case called for a much smaller space; basically, a space about 10 feet by 10 feet while the puppeteer stands in front of a computer desk.

Removing motion capture products that required a large footprint left companies that were using inertial motion technology. These are sensor-based systems that don't use cameras to track the body but the position and motion of the body itself [17]. Only one option met the price goals, so it was pursued for further research.

The integration is comparable to face tracking technology. There were a few dozen sensors on the body all sending position, orientation, and velocity information over the network at a frequency matching each render frame. By connecting those sensor points to points on our character's skeleton to drive the in-game position, orientation, and velocity, it resulted in the ability to run both the facial capture server and the body capture server on the same machine. Frame rates remained sufficiently high to see positive results.

Initially the out of the box solution was used to attach the body sensors. A chest mount was used to attach the camera. The chest mount was developed in the laboratory as way to keep the camera pointing at the actor's face. The combination of all the straps, wires, mounts, and batteries made it a bit cumbersome for the actor. The chest mount was not as stable as hoped, and interfered with arm and hand motions (See Fig. 1).



Fig. 1. Chest-Mount prototype with sensor straps

To pursue a more practical use case, the solution moved away from the chest mount and explored head mounted, or helmet solutions. The initial prototype included a low-cost bike helmet with GoPro mounting hardware created within a couple days. To alleviate some of the cumbersome nature of the straps on the body, a motion-capture suit was used. Sensors were attached using hook and loop fastener tape attached to the suit jacket and pants. This simplified preparation for the actor (See Fig. 2).



Fig. 2. Head-mounted prototype with sensors on motion capture suit

The combination of motion-capture suit and camera mounted on the bike helmet became our first successful puppeteering prototype. This is the version we widely demonstrated and documented [1]. The Defense Equal Opportunity Management Institute (DEOMI) saw this prototype and funded additional work to improve the prototype. The team worked with their researchers to develop research studies on learning topics best supported by the technology.

With the new prototype the team began exploring training scenarios that would benefit from this technology. The specific use case involved a group setting with

co-facilitators as the trainees. The puppeteer could jump into the body of various group participants, expressing full human emotion in each role. Despite the plan for the characters to stay seated in the group session, the team created strategies to allow puppeted characters to walk using a one-handed controller. The character can move with a walk animation while still being able to control the face, head, arms and hands. This allowed for realistic interactions such as walking down the street with someone gesturing to various points of interest along the way.

Test #5: Test Facial Tracking with New Vendor. During the development process, the company that developed the product we were using for the facial capture was purchased by Apple [18]. Once again, options were limited for a replacement, but the team persisted to make use of a capability provided by a large commercial company known for facial animation in video games with a real-time component.

This facial capture product was a bit different in a few key ways: 1. It makes use of a standard webcam rather than a depth camera; 2. The facial shapes were different, meaning we had to edit all the current character models; 3. Finally, it allowed for quick calibration, meaning it could define the position and outline of the mouth, nose, and eyes. It was not necessary to calibrate each unique face and facial expression.

At first, the results were somewhat less impressive than what we had with the previous vendor. Lip syncing during speech was as fluid as the previous product, but it was more difficult to achieve key emotions that involve smiles, frowns, and furrowed brows. Software updates, such as one that added shape modifiers, have allowed the team to tune a face shape to be prominent. For example, an actor's smile can be magnified to be more obvious on the in-game character.

There were two more issues that needed to be addressed during this phase of research. Low camera frame-rate caused delays in seeing expressions in real-time. Market research indicated a paucity of traditional webcams that support 60 frames per second update rate. Thankfully, an eponymous (GoPro-like) camera with high resolution and framerate was available but a video capture card was required for the computer to treat the feed like a webcam feed.

The second issue was the helmet design. The bike helmet was not stable; moving around on the head depending on the size of the actor's head. The helmet was heavy and the additional weight of the camera attached to it pulled the front of the helmet downward. This often caused the helmet to obscure the forehead and brows. Next the team tried using a military tactical helmet. This type of helmet is built to have attachments such as a flashlight connected. Unfortunately, this helmet was even heavier and more cumbersome than the bike helmet. The team settled on using headgear used by wrestlers. This headgear is made up of two solid pieces covering the ears, two straps overhead, and one strap behind the head (See Fig. 3). They are lightweight and can be adjusted to fit any head size. The team's Art Director modeled and 3D printed hardware to mount the camera onto the helmet. This solution, along with lightweight carbon fiber rods, provides exactly the form, fit and function needed.

Test #6: Evaluate New Technology in Real-Time Body Tracking. Throughout this process, the team has experienced elation and frustration. Emerging technology does not always work the way we would like. For example, the body tracking system was frequently plagued with two issues; drift and inaccuracy. Because the motion sensor



Fig. 3. Lightweight adjustable headgear with camera and light assembly

technology is based on magnetic sensors, the sensors drift over time. This means that visual anomalies occur, such as arms starting to clip through parts of the body, or the position of the body in space might shift from center (See Fig. 4). If the actor is sitting on a metal chair, the metal or any wire can create interference causing the virtual body parts to move or be positioned in unnatural ways.



Fig. 4. Character showing some Clipping of Hands

Aside from the issue with drift, there is also an issue with general inaccuracy. Calibration occurs only for specific poses. Even after calibration, arms may not appear in a natural position at the sides of the body. However, the technology does not allow an actor to put the palms of their hands together. Hands are not described as collision bodies, nor do they have “sticky surfaces” that attract the hands together. Rather, while the actor’s hands are touching, the character might appear to have their hands a foot

apart or crossed over one another, clipping into one another. This is true for both the arms and legs. A good actor could work around these issues and working around the system limitations, but the intent was that movement would be natural so that anyone could play the role of the puppeteer.

4 Outcomes

Though the research area described above is in its infancy, the outcomes show great promise. The resulting prototype has functionality that allows an actor to select a virtual character in a scene and take over that character's control. The actor expresses natural facial expressions and body movements that are played out by the virtual character. The facial tracking system and the body tracking systems are working in conjunction with lower body movement and idle animations. For example, if the actor guides the character forward, using a one-handed controller, the walking animation begins with the lower body, but the actor is able to point and make facial expressions at the same time. This capability goes far in providing a prototype for experimentation.

5 Way Ahead

The prototype is being used to support a wide range of research studies. These will be the topic of future papers. The team continues to do market research into technical advancements and market shifts. Real-time motion capture is a fledgling technology, so no turn-key solutions are currently available. There are many vendors emerging in inertial motion sensors, and this team looks forward to evaluating them to see if there is a solution to the drift and inaccuracy issues described above.

One company has a solution that is showing promise. Their product uses the HTC Vive (See Fig. 5) virtual platform along with inverse kinematics software to achieve real-time motion capture with fewer sensors [19].

The setup involves using the lighthouse (sensors on a stand that work with the HTC Vive) and sensors from the Vive. The Vive sensors track very accurately through the Vive lighthouses. By wearing sensors on both feet, at the waist, on top of the helmet, at the elbows, and while holding the Vive controllers, the entire body can be tracked and moved in real-time (See Fig. 5). Inverse kinematics is software that interpolates the locations of points on the body that cannot be tracked based on the body position of the sensors that are being tracked [19].

The team is building a community of practice on the topic area. The goal is to have this community meet on a yearly basis at the HCII conference to discuss the status of the state-of-the-art and demonstrate progress.



Fig. 5. IKinema's orion full body MoCap on Vive [19]

6 Conclusion

The prototype development described in this paper is intended to build a community of practice interested in building low-cost real-time motion capture capabilities. The development of this prototype, from a game controller feeding specific, clean, animations to a character to the real-time body and facial tracking solution that exists today is useful in informing others with similar interests. As new solutions come to market this team will continue to monitor and evaluate market solutions, providing the results in papers and at conferences as appropriate. The community-of-practice on the topic area is expected to build on this team's research. As this technology area evolves, it is expected that both this team and the community will continue to share status of the state-of-the-art and be able to demonstrate continual progress.

Acknowledgements. The ARL STTC, Cole Engineering Services, Inc., and the Institute for Simulation and Training and the University of Central Florida would like to express our appreciation for the support of the Defense Equal Opportunity Management Institute for resources and expertise provided to this technology area.

References

1. Griffith, T., Dwyer, T., Kinard, C., Flynn, J.R., Kirazian, V.: Research on the use of puppeteering to improve realism in army simulations and training games. In: Lackey, S., Shumaker, R. (eds.) VAMR 2016. LNCS, vol. 9740, pp. 386–396. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39907-2_37
2. Bailey, K.: <http://www.techcrunch.com>, <https://techcrunch.com/2017/02/25/conversational-ai-and-the-road-ahead/>. Accessed 25 Feb 2017
3. Merritt, C.: <http://www.towardsdatascience.com>, <https://towardsdatascience.com/a-driver-less-world-c3baea61734b>. Accessed 27 Sept 2017
4. Summers, N.: The real-time motion capture behind ‘Hellblade’. How a tiny team in Cambridge, England, brought Senua to life. <https://www.engadget.com/2017/08/08/ninja-theory-hellblade-motion-capture-demo-video/>. Accessed 8 Aug 2017
5. Peel, J.: www.pcgamesn.com, <https://www.pcgamesn.com/hellblade-senuas-sacrifice/hellblade-unreal-engine-4>. Accessed 27 Nov 2017
6. Boyle, E.: Tech Crunch. <http://www.techradar.com/news/this-is-the-tech-that-will-make-games-look-better-and-cost-less>. Accessed 8 Aug 2017
7. Souppouris, A.: <http://www.Engadget.com>, <https://www.engadget.com/2015/09/16/ninja-theory-hellblade-indie-game>. Accessed 16 Sept 2015
8. Hanson, B.: www.gameinformer.com, <http://www.gameinformer.com/b/features/archive/2017/04/12/watch-us-push-hellblades-mocap-process-to-its-limits.aspx>. Accessed 12 Apr 2017
9. Ninja Theory: Hellblade: Diary 21 - Making a Virtual Human | Real-time performance capture. <https://www.youtube.com/watch?v=CmrXK4fNOEo&index=21&list=PLbpf8TRYizaT6GfMcKBG-RoUOQ6BJRXP>. Accessed 17 Mar 2016
10. Maxwell, D.B., Griffith, T.S., Finkelstein, N.M.: Use of virtual worlds in the military services as part of blended learning strategy. In: Handbook of Virtual Environments, pp. 959–1000. CRC Press-Taylor & Francis, Boca Raton (2014)
11. Griffith, T.S., Ablanado, J., Nenneman, M.: Taking it to the EDGE. In: Proceedings of the International Defense and Homeland Security Simulation Workshop, Bergeggi, Italy (2015)
12. Moore, M.E., Novak, J.: Game Industry Career Guide. Cengage Learning, Delmar (2010)
13. Glanville, S.: Object editor. In: Anim8 or Manual (2006)
14. Brier, D.: E3: New info on Microsoft’s Natal – how it works, multiplayer and PC versions. http://old.seattletimes.com/html/technologbrierdudleysblog/2009296568_e3_new_info_on_microsofts_nata.html. Accessed 3 June 2009
15. Marquand, R.: Star Wars Episode VI: Return of the Jedi; George Lucas Commentary (Film). 20th Century Fox (2004)
16. da Silva, D.B., Vidal, C.A., Cavalcante-Neto, J.B., Pessoa, I.N.S., Nunes, R.F.: A robust balance strategy applied to real-time animation data with Kinect sensor. In: 2017 19th Symposium on Virtual and Augmented Reality (SVR), pp. 8–17 (2017)
17. Originating Technology/NASA Contribution: Inertial Motion-Tracking Technology for Virtual 3-D (2005). https://spinoff.nasa.gov/Spinoff2005/ch_10.html
18. Baraniuk, C.: Faceshift: Apple buys Star Wars motion-capture company. BBC News. <http://www.bbc.com/news/technology-34920548>. Accessed 25 Nov 2015
19. Feltham, J.: <http://www.Uploadvr.com>, <https://uploadvr.com/ikinema-orion/>. Accessed 23 Mar 2017
20. McAlone, N.: See the amazing tech behind the original Jabba the Hutt puppet - which managed to fit 3 people inside of it. <http://www.businessinsider.com/the-tech-of-the-jabba-the-hutt-puppet-2015-11>. Accessed 10 Nov 2015
21. Ninja Theory: Hellblade Development Diary 5: Business of Creation (2014). https://www.youtube.com/watch?v=_f3NMMrQFEE