Peer-Sourced Media Management for Augmented Reality

Raimund K. Ege
Dept. of Computer Science
Northern Illinois University
DeKalb, IL, USA
email: ege@niu.edu

***Abstract*—Wearable devices with advanced recording devices enable the capture of current scenes and scenery in real time. It enables device holders to become media producers. Moreover, with its computing power and network connectivity, the wearable device can become a peer in a peer-to-peer based content delivery network. In this paper, we will describe a framework for allowing peers to join a content capture and delivery system that collects real-time media streams with virtual reality models in the cloud. The combined media pool represents an augmented reality world which is made available back to the peers and rendered onto their wearable devices. Issues arise, such as the combining and correlating of media streams in real time as well as capturing reference data from related streams and virtual reality models: we discuss our approach. We also outline an implementation in the Java programming language for Android and cloud platforms to justify and demonstrate the feasibility of our approach.**

Keywords-Android; augmented reality; virtual reality; peer-to-peer systems; multi-media content delivery

I. INTRODUCTION

Wearable devices with computing power, display and recording capabilities are becoming increasingly available and affordable. Moreover, these devices are network connected at high speeds, low latency and high bandwidth. In this paper, we will describe a framework for pairing these smart devices with computing power from the cloud to form a peer-sourced media content management system that enables realistic immersion into an augmented reality world.

A real-world application scenario could be a team of firefighters entering a burning building on a search and rescue mission. Each firefighter is equipped with network-connected sensors that capture video, audio, temperature, air quality data etc. The multiple streams of data are gathered and combined by cloud-based compute nodes with a virtual reality model of the building to form an augmented reality model. Each firefighter wears a network-connected display device in his/her helmet which receives a customized heads-up display of his/her forward view and situation. Even if smoke has filled the immediate surroundings of the firefighter the heads-up display might enable him/her to accomplish life-saving actions.

The framework we describe in the paper has many components, which we will describe in detail. Section 2 surveys the state of research as it relates to virtual and augmented reality, wearable devices and multi-media mediation. Section 3 explains how we gather multi-media sources from sensors and tag them suitable for adequate correlation and mediation. Section 4 details how peers are established, join the content delivery network and establish trust relationships with each other. Section 5 covers how tagged media streams are embedded into a reference virtual reality model and rendered into a geo-referenced attitudinal output media stream suitable for a heads-up display. Section 6 reports on the status of our prototype implementation in Java for cloud-based compute nodes and Android-based handheld and wearable devices. We conclude the paper with an assessment of our approach and future directions for our research.

II. BACKGROUND

In the computer and game console world, multi-player games are common: players throughout the Internet participate in a fantasy world and interact for a purpose, typically chasing and fighting enemies and each other in real time. Such multi-player games are part of the larger augmented virtual reality set of applications with a long research history [1]. These applications are being investigated for uses in telemedicine, manufacturing, etc. The basic idea is to combine a virtual model with actual sensor data to guide humans in their endeavor. The advent of wearable devices with a wide range of sensors is enabling a richer and deeper immersion in the given scenario [2] [3]. The ultimate goal is to increase the ratio of actual "real" content to virtual content. Reducing the virtual reality component to zero yields pure augmented reality.

The key capability of an augmented reality system must be the accurate recording from a real-time sensor. Multi-media I/O components include high-definition screens and video cameras, high-fidelity speakers and microphones. Plus components to determine device location, position, and attitude: GPS, accelerometers, compass, etc. Not only must the data be recorded and made available in real-time, but it must also be annotated and tagged with a variety of attributes to enable correlation with other streams and embedding into a reference frame work [4]. For example, a video recording stream, in addition to capturing the sequence of video frames must also record exact time and location where the location must include direction and attitude. The richness of such metadata determines how precise and life-like the resulting augmented reality world will be [5].

Wearable connected computing devices, such as wristwatches and even eye glasses (Google GLASS) have reached the consumer market. The focus is shifting from computing and storage capabilities on these devices to connectivity and multi-media sensor and reproduction components. Connectivity capabilities are typically wireless and include high-bandwidth cellular (4G, LTE) and WLAN (IEEE 802.11) connections, plus lower-bandwidth near field connections (Bluetooth, NFC, etc.). Transmission rates in the multi megabits per second range and latency rates in the sub millisecond range are currently quite standard.

Who does the recording and contributes is another important aspect of sourcing sensor data to augment reality. Is the other player that appears in a shoot-first-ask-questions-later game a friend or foe? In peer-sourced augmented reality systems, the management of the multi-media source and establishment of trust is essential. In our prior work [6] [7], we investigated the authentication of participants in peer-to-peer networks, the establishment and management of trust, and the use of such media sources in building content management systems. An important lesson was that while modern mobile devices are compute-capable, cloud-based components add additional heft and authority to a seamless and smooth creation of a truly immersing virtual and augmented reality experience.

III. Media Capture

It all starts with recording something in real time using a sensor. The type of sensor can be a video or audio recorder, a location sensor, an attitude sensor, or even a sensor of biological data, and many more. In addition to the actual data sensed, it must be packaged with the exact time of recording. Multiple streams of sensor data are combined into a multi-media stream which interleaves its content streams plus provides meta-data to ensure their proper sequencing and correlation. It is important that the container format used to wrap the content streams is flexible enough to accommodate not only the stream data but also extensive amounts of reference information used to combine the streams. We are using an extension of the WebM project [8] format. The WebM container format is an open standard and allows us to collate an unlimited number of video, audio, pictures and subtitle tracks into one stream. We add the capability of identifying reference elements at identified points in time and at locations.

Video data is the key stream type captured via video sensors, i.e., cameras, available on the wearable devices carried by a peer. Video is captured as a sequence of video frames. Each frame carries a time stamp as major meta reference data. Equally important is the location of video capture, lens parameters and attitude, i.e., which way the camera points. Our container format allows us to group sequential video frames into video sequences that share a common location. We represent the location with a “Normal Vector”.

Figure 1. Normal Vector

Figure 1 shows how a normal vector captures not only the location of a video plane but also its relative position. The normal vector is represented via 2 points: its origin and extent points. Both points are captured in absolute latitude and longitude coordinates. While the distance between origin and extent point of the normal vector is not normally relevant, we use the length of the vector as a guide to the size of the video frames being referenced. A longer vector indicates a larger area shown in the video. We use the length of the normal vector when attaching multiple streams into a virtual reality frame.

Audio data is captured by microphones and sequences into frames that are referenced with a time stamp. While it would be possible to also capture and store directional information, which might be meaningful in the case of a directional microphone, we are able to deduce that information from the normal vector stored for video frames recorded at the same time on the same device. Of course, if the wearable device only records video, then such directional information is not available. We are considering this extension for future work. The audio data with its correlated reference metadata is also wrapped into the same container as the video data.

Any other data, such as gathered from biological data sensor, is equally framed and referenced. Examples of such data might be the heart rate of the person wearing the device, the temperature of the surroundings, or movement/acceleration data measured. Our container format allows a free-form type designator that enables sensors of any kind, as long as their sensed data can be digitized and framed.

Our container format also allows the carrying of virtual reality model data. Actually, such data is similar in nature to "real" data, but is derived not from sensors but from virtual reality models of the surroundings that the wearers of the wearable devices inhabit. To allow clear distinction of sub-streams within the container, each sub-stream carries a unique stream identifier, which is correlated to a stream dictionary that holds relevant information about the sub-stream. The complete stream dictionary is embedded into the multi-media stream at regular intervals.

IV. Peer Management

The richness of the resulting augmented reality depends on the number of contributing peers. Users with wearable devices can join the network and become peers in the peer-to-peer content sharing network. In our prior work [5, 6] we investigated the authentication of participants in peer-to-peer networks, the establishment and management of trust, and the use of such media sources in building content management systems. In peer-sourced augmented reality systems management of multi-media source and establishment of trust is essential.

Our approach delegates peer identification to an OpenId provider, but maintains a shared understanding of how trust-worthy a peer is. Peers that have reached a trust threshold – initially just one bootstrap peer – maintain a database of trust information (called trust nuggets) per peer. The trust nuggets are encrypted with a private key that is shared by all trusted peers. The trust nugget stores information about a peer's past participation in the content sharing network and exposes the peer's trust rating. The public key to decrypt the trust nugget is shared among all peers that participate in the network, which enables any peer to evaluate another peer's trust worthiness.

Each peer can produce and consume media streams. A stream produced by a peer carries the peer's trust value. And a peer is only allowed to consume streams that match his/her trust value. Each successful participation of a peer in a construction of an augmented realty scenario adds to the peer's trust value. The necessary adjustment of the trust nugget has to be performed by a trusted peer, who will then reseal the trust nugget with the private key.

Each peer also generates a public/private key pair. The public key is also stored in the peer's trust nugget. The public/private key pair is used when peers exchange streams: the key pairs are used to establish a shared session key, which is used to encrypt and decrypt the streams.

V. Media Mediation

Media streams that are collected by wearable devices worn by peers are embedded into a reference virtual reality model and rendered into a geo-referenced attitudinal output media stream suitable for a heads-up display. The key to an exact construction of a resulting stream is that input streams are mediated into a stable reference model. This model is provided by a virtual reality model of the surroundings of the scenario that the peers inhabit. We use the Virtual Reality Modeling Language (VRML). Geo-referenced meta and attitudinal data that accompanies the peer-gathered streams is used to create an augmented reality model. The resulting augmented reality model is rendered into a resulting media stream which is available to participating peers.

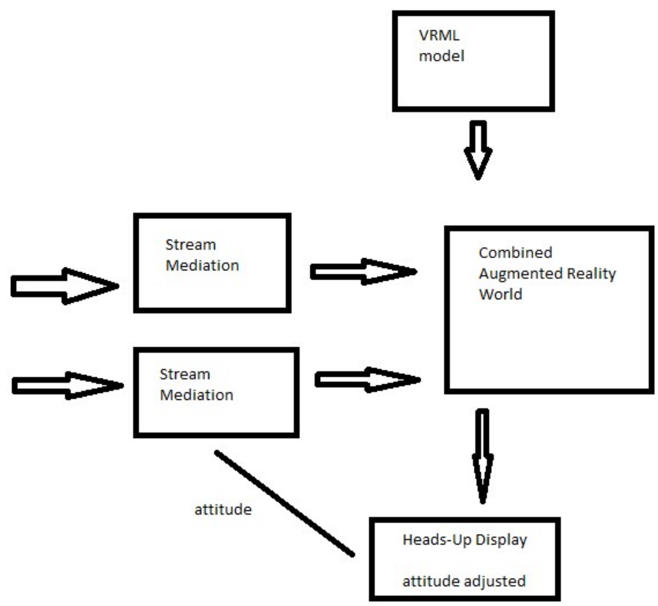

Figure 2. Cloud Stream Mediation

Figure 2 shows how peer-contributed media streams are combined and embedded into a reference world that is created from input using the VRML standard. The mediation of input streams can involve geometric conversions to adjust the spatial location and dimensions to produce a life-like presentation. The rendering of the resulting augmented reality can be adjusted to conform to the attitude of an input stream: this enables a peer to view the resulting stream in the same geo space as its own recording device.

The stream mediation is performed by a cloud-based general purpose peer. While any peer could perform this peer functionality, it requires significant computing power. Modern wearable devices are gaining computing power as technology progresses, but making this capability available in the cloud allows any peer to participate without risking to hamper the usability of its device.

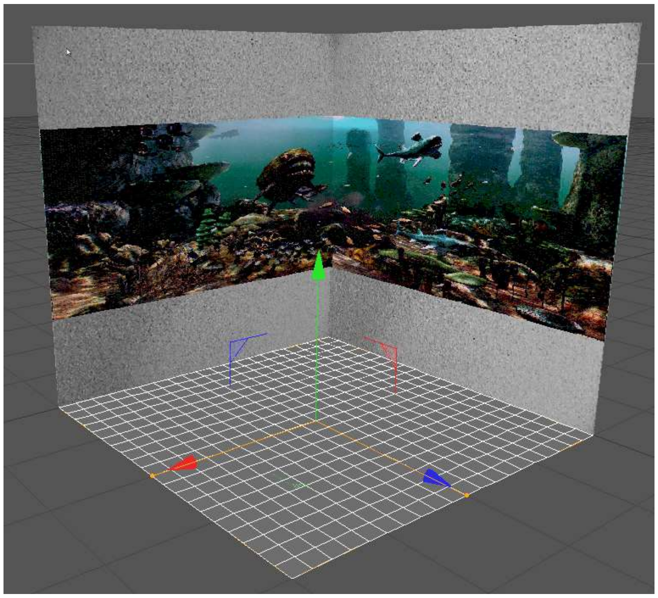

Figure 3. Augmented Reality World

Figure 3 shows a simple scenario of a cube-shaped world, where 2 input streams are embedded onto walls. The streams are contributed from two peers, the cube context is provided in VRML format.

VI. PROTOTYPE

Our effort includes the implementation of a prototype to demonstrate the feasibility of our approach. We provide peer capabilities in a modularized fashion: each module provides a feature that represents typical peer functionality. An actual peer can be constructed by assembly from one or more modules. All modules are programmed in the Java programming language suitable to be applied into an app for the Android platform. Peers can also live in the cloud, all they need is a Java virtual machine to run.

Four modules are part of our prototype implementation: (1) peer management: it implements peer authentication, key generation and heart beat connections to the trusted peer in the network; (2) source stream creation: it implements the selection of suitable input sensors, the recording a data from the sensor, the collection of meta data and the creating of the container stream to be output from this peer; (3) stream consumption: it implements the selection of an input stream and its display on an output devices; and (4) stream mediation: it implements the adaptation of input streams into a virtual reality world to form a augmented reality world. The peer management module also includes the capabilities of the trusted peer once the peer has reached the threshold required.

A typical peer app will include peer management, source stream creation, and stream consumption to enable a peer to be an active participant in an augmented reality scenario. The peer contributes the stream captured by its own sensor, e.g., video, and displays the combined and mediated resulting world from a cloud-based source.

VII. CONCLUSION

In this paper, we described a framework whose components produce a life-like augmented reality world. We gather multi-media sources from sensors and tag them suitable for adequate correlation and mediation. Peers join a content delivery network and establish trust relationships among each other. Cloud-based media streams – mediated based on their associated metadata - are embedded into a reference virtual reality model and rendered into a geo-referenced attitudinal output media stream suitable for a heads-up display. We are working on a prototype implementation in Java for cloud-based compute nodes and Android-based handheld and wearable devices.

Our goal for future research is to enhance the media mediation capabilities of our cloud-based peer components. We envision a rich augmented reality world that is populated by many dynamic input streams. While our current approach only provides one output stream that is adjusted from one input stream's geo attitude, we will work on better feedback from a given peer's input to the output that is consumed by the same peer.

REFERENCES

[1] R. Azuma, et al., "Recent Advances in Augmented Reality," IEEE Computer Graphics and Applications (CGA) 21(6):2001, pp. 34-47.

[2] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond and D. Schmalstieg, "Real-Time Detection and Tracking for Augmented Reality on Mobile Phones," IEEE Transactions on Visualization and Computer Graphics, 16(3), 2010, pp. 355-368.

[3] A. Morrison, et al., "Collaborative use of mobile augmented reality with paper maps," Journal on Computers & Graphics (Elsevier), 35(4), 2011, pp. 789-799.

[4] E. Macias, J. Lloret, A. Suarez and M. Garcia, "Architecture and Protocol of a Semantic System Designed for Video Tagging with Sensor Data in Mobile Devices," Sensors,vol.12, no. 2, 2012, pp. 2062–2087.

[5] Meenakshi Sundaram V., Shriram K. Vasudevan, A. Ritesh and C. Santhosh, "An Innovative App with for Location Finding with Augmented Reality using

CLOUD," 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), Procedia Computer Science 50, 2015, pp. 585 – 589.

[6] Raimund K. Ege, "Peer to Peer Media Management for Augmented Reality," International Conference on Networking and Services (ICNS 2015), Rome, Italy, May 2015, pp. 95-100.

[7] Raimund K. Ege, "Secure Trust Management for the Android Platform," International Conference on Systems (ICONS 2013), Seville, Spain, January 2013, pp. 98-103.

[8] The WebM Project - About WebM. http://www.webmproject.org. [accessed August 19, 2015]