VR Swarms: Enabling Shared Virtual Experiences

Muhammad Huzaifa
huzaifa2@illinois.edu
University of Illinois at Urbana-Champaign

Abdulrahman Mahmoud
amahmou2@illinois.edu
University of Illinois at Urbana-Champaign

Abstract

We present the notion of *VR swarms*, where groups of users collaborate to create an immersive, shared environment. We present several interesting applications enabled by this concept, an approach to implementing it, and highlight key areas of research driven by it.

1 Introduction

Over the past few decades, the state-of-the-art in computing applications has advanced very rapidly. Today, people of all ages and lifestyles have the equivalent of a supercomputer available in their pockets, with far more complex applications compared to the number-crunching usage of yesteryear. All these advances have ushered us into an era where virtual reality (VR) is no longer an object of sci-fi imagination, but rather a real-world technology with immense potential across many facets of our lives. This paper discusses the concept of VR "swarms" [2], wherein a group of physically nearby users creates a collaborative, immersive, and shared virtual environment which can enhance physical reality as we know it. We discuss some exciting applications, technical approaches, and various challenges and research opportunities created by this line of work. We use the term VR as an umbrella term for Augmented/Virtual/Mixed/Extended Reality (AR/VR/MR/XR) throughout the rest of the paper.

2 Applications

Although VR swarms will enable a multitude of applications, we discuss three representative ones here.

Imagine an arbitrarily large physical space full of real people, where each person's VR headset is equipped with speakers and omnidirectional microphones. By linking all the speakers in the room, it would be possible to recreate realistic 3D audio in that physical space. Each channel of the incoming audio signal would be mapped to the headsets whose physical locations best correspond to the directions of the incoming sound. Such a setup could recreate live concerts, speeches, and even the feeling of standing in a stadium during the Super Bowl. Extending this further, by creating a virtual microphone array consisting of all the physical microphones, the sounds in the room could be captured with extremely high fidelity and spatial localization. Doing so would enable *immersive telepresence*, where two or more physically separated swarms will be able to share a virtual environment. The microphones would record sound on one end and the distributed speakers would recreate that sound on the other end. Crucially, people would be free to roam around – the system will track each user's location in real time and recompute the sound-to-user mapping on the fly.

VR swarms would have a significant impact on entertainment as well. Be it playing virtual soccer with gravity-defying soccer balls, dunking basketballs with Michael Jordan, playing otherwise unplayable games such as Quidditch, or recreating the Battle of Waterloo with your friends, VR swarms would redefine modern entertainment.

Finally, one of the most important applications of VR swarms would be *dress rehearsals*. Firefighters could be trained by presenting them with virtual wildfires, army doctors could be given field training by placing them in the middle of virtual battlefields, anti-riot police could be trained by confronting them with virtual riots, and humanitarian relief efforts could be optimized by immersing the staff in a virtual target zone. In all these scenarios, each user will be in a shared physical space with other users, where the space would consist of a mixture of virtual (fires, stretchers, food, etc.) and real objects (the ground, other people, equipment, etc.). As virtual objects interact with physical ones, AR lenses would visualize the interactions, speakers would auralize the interactions, and haptics would physically simulate the interactions [1] (unfortunately, this paper is neither wild nor crazy enough to even remotely talk about gustation and olfaction).

3 Approach

While VR swarms could be implemented in a centralized fashion where one dedicated node runs the entire application by taking in information from other swarm users, such an approach would not be feasible for several reasons. First, a centralized approach would have high communication latencies, making it difficult to achieve real-time deadlines. Second, a centralized scheme would not scale to a large number of users (e.g., 11 vs 11 in virtual soccer). Third, communication between far-away nodes (e.g., users hundreds of feet away in a field) would require power-hungry antennas, which would not be possible given the energy constraints. Finally, given the ever-changing positions of users, choosing the optimal server may not possible.

Therefore, we propose a distributed architecture where each user would contribute to the overall computation. Each user would share local information with all neighboring users, creating an extremely distributed and communication-centric architecture. For instance, each user would run a local SLAM (Simultaneous Localization and Mapping) [9, 12], exchange local maps with its neighbors, and then stitch them all together to obtain a global picture of the environment. Similarly, each user would render only a portion of the virtual environment,

[1] This would require each user to wear a haptic feedback suit.

and then would use the renderings from its neighbors to create the entire environment. This scheme would also apply to distributed audio and recording, and perhaps even shared haptics, where a multi-user, multi-virtual-object interaction might necessitate exchanging haptic feedback information (e.g., two firefighters colliding with each other and a virtual tree simultaneously).

4 Challenges and Research Opportunities

Designing such a system would require innovations throughout the computing stack, all the way from algorithms to hardware. Here, we highlight some of the fundamental challenges that would need to be solved at each layer.

Applications: Developing applications for a system where essentially all computation is decentralized would require a drastic change in the way programmers think about algorithms. A local-compute-and-multicast paradigm would require reasoning about things such as computation handoff, where a particular process or job or computation is transferred from one user to another in real-time. For instance, the rendering of a fire hose as it gets passed from one user to another. Another algorithmic consideration would be the use of approximate computing for pruning out unnecessary computations. An example would be *distance-based* computation: based on the distance from the user, the accuracy of SLAM could be increased to centimeters or decreased to inches, resolution could be increased to fovea level fidelity or decreased to 4K [10], and work pertaining to objects outside a user's field of view could be culled. Finally, programmers would have to think about optimizing global performance as opposed to local performance.

Operating Systems: Swarm VR would require a real-time OS, with tight, but soft real-time deadlines. Achieving such deadlines would require lightweight primitives for processes, threads, and most importantly, the communication stack. With communication a first-class citizen, short range communication protocols (perhaps derived from IEEE 802.11p [4]), APIs, system calls, drivers, and firmware would all have to be specialized to eliminate overheads of redundant data copying and extraneous interfaces. An existing example of such an optimization is the dedicated VR drivers in Windows that do not treat HMDs as traditional displays [8], and therefore improve performance [2].

Programming Languages: While programmers are already proficient at using MPI, Spark, or Storm for writing distributed applications, distributed graphics is an example of an application where current programming abstractions are insufficient. For instance, to support computation handoff, abstractions for describing shared, distributed objects would helpful. For perception-driven rendering [11], support would be required for tagging objects with approximation permissions; i.e., whether an object can be approximated or not. For instance, in a riot, the texture of a far away virtual person's shirt may not matter, but any harmful objects they may be carrying would need to be rendered with high-fidelity at all times due to their importance. Finally, techniques would be required for the validation and verification of such programs.

Compilers and Runtimes: The rich information provided by a new programming language would be used by compilers and runtimes to pass on higher-level semantics to the hardware, such as the priority of a particular piece of data (for communication purposes), source of data (local or remote to dictate caching policies), level of approximation, etc. Furthermore, compilers would require a new IR for describing distributed computations and their interactions, enabling more optimizations in the process.

Hardware: In addition to the challenges faced by "traditional" VR – high resolution, frame rate, field-of-view [3, 7], low latency [5], area, power, and small form factor [6] – swarm VR hardware would need to solve several new challenges. First and foremost, the OS stack alone would not be able to solve the communication latency problem. Hardware support would be needed for short range communication protocols. This support could be in the form of integrating I/O (network, sensors, USB ports, ADCs) into unified, shared memory, thereby removing extraneous copies and explicit software management overhead; or, if we may be so bold, it could be in the form of *swarm-wide unified, shared memory*, where coherence protocols across the entire ad-hoc network would make communication implicit. Second, a new ISA would be required to enable the compiler or runtime to pass down meaningful information. The ISA could provide macro-instructions for communication and data sharing. Third, accelerators would be required, as is currently assumed by default, but with far more stringent area and energy constraints – small SoCs with hours of battery life without discomfort under high computational load. Finally, efficient and small form-factor batteries, mechanical devices, and fabrics would be required to support realistic haptics.

Security and Privacy: As cameras, microphones, and other sensors will be present in all VR headsets, security and privacy will be of paramount importance. Mechanisms will be required to ensure a user's privacy is not violated, especially involuntarily by other users [1]. For instance, users could broadcast their privacy policies (e.g., okay to share silhouette; not okay to share facial features) to nearby users. Techniques will also be required to ensure that the swarm-wide communication paradigm is not hijacked to carry out security attacks.

5 Conclusion

We presented the idea of a *VR swarm*, where users collaborate to create a shared, immersive environment. VR swarms can enable a plethora of interesting applications, ranging from telepresence to emulating battlefields. We also outlined several research challenges associated with this idea. Perhaps most of them are many, many years away from being solved but we hope that they will provide research directions and food for thought for the readers.

[2] They are still not fast enough for high framerate rendering.

Acknowledgements

This material is based upon work supported by the Applications Driving Architectures (ADA) Research Center, a JUMP Center co-sponsored by SRC and DARPA.

References

[1] Tousif Ahmed, Apu Kapadia, Venkatesh Potluri, and Manohar Swaminathan. 2018. Up to a Limit?: Privacy Concerns of Bystanders and Their Willingness to Share Additional Information with Visually Impaired Users of Assistive Technologies. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 89 (Sept. 2018), 27 pages. https://doi.org/10.1145/3264899

[2] Levent Bayindir. 2016. A Review of Swarm Robotics Tasks. *Neurocomputing* 172 (2016), 292 – 321. https://doi.org/10.1016/j.neucom.2015.05.116

[3] David Kanter. 2012. Graphics Processing Requirements For Enabling Immersive VR. Website. http://developer.amd.com/wordpress/media/2012/10/gr_proc_req_for_enabling_immer_VR.pdf

[4] Daniel Jiang and Luca Delgrossi. 2008. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*. 2036–2040. https://doi.org/10.1109/VETECS.2008.458

[5] John Carmack. 2013. Latency Mitigation Strategies. Website. https://danluu.com/latency-mitigation/

[6] Andrew Maimone, Andreas Georgiou, and Joel S. Kollin. 2017. Holographic Near-eye Displays for Virtual and Augmented Reality. *ACM Trans. Graph.* 36, 4, Article 85 (July 2017), 16 pages. https://doi.org/10.1145/3072959.3073624

[7] Michael Abrash. 2014. What VR Could, Should, and Almost Certainly Will Be Within Two Years. Website. http://media.steampowered.com/apps/abrashblog/Abrash%20Dev%20Days%202014.pdf

[8] Microsoft. 2018. EDID Extension (VSDB) for HMDs and Specialized Displays. Website. (2018). https://docs.microsoft.com/en-us/windows-hardware/drivers/display/specialized-monitors-edid-extension

[9] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. 2017. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications* 9 (12 2017). https://doi.org/10.1186/s41074-017-0027-2

[10] Martin Weier, Thorsten Roth, André Hinkenjann, and Philipp Slusallek. 2018. Foveated Depth-of-Field Filtering in Head-Mounted Displays. *ACM Trans. Appl. Percept.* 15, 4, Article 26 (Sept. 2018), 14 pages. https://doi.org/10.1145/3238301

[11] Martin Weier, Michael Stengel, Thorsten Roth, Piotr Didyk, Elmar Eisemann, Martin Eisemann, Steve Grogorick, Andre Hinkenjann, Ernst Kruijff, Marcus Magnor, Karol Myszkowski, and Philipp Slusallek. 2017. Perception-driven Accelerated Rendering. *Comput. Graph. Forum* 36, 2 (May 2017), 611–643. https://doi.org/10.1111/cgf.13150

[12] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* 35 (09 2016). https://doi.org/10.1177/0278364916669237