Mica: A Photoreal Character for Spatial Computing

James Bancroft
Nafees Bin Zafar
Sean Comer
Takashi Kuribayashi
Jonathan Litt
Thomas Miller

Magic Leap Inc.

CCS CONCEPTS

• **Computing methodologies** → **Procedural animation**; **Mixed / augmented reality**.

ACM Reference Format:
James Bancroft, Nafees Bin Zafar, Sean Comer, Takashi Kuribayashi, Jonathan Litt, and Thomas Miller. 2019. Mica: A Photoreal Character for Spatial Computing. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Mica is an autonomous, photoreal human character that runs on the Magic Leap One spatial computing platform. The past ten years have seen tremendous improvements in virtual character technology on screen, in film and games. Spatial computing hardware such as Magic Leap One allows us to take the next step: interacting with virtual characters off the screen. We believe that virtual characters who are aware of and can interact directly with users in real-world environments will provide a uniquely compelling experience.

Mica runs under limited hardware constraints relative to recent work in the field of high-end, real-time virtual characters [Games 2019; Seymour 2018]. The performance capabilities of Magic Leap One are significant for a battery-powered device but are naturally less than a desktop workstation that uses orders of magnitude more power. At the same time, there is a high bar that must be met to create a convincing virtual character; rendering, animation, deformations, clothing, hair, and behavior must all meet a certain intangible quality level for the character to be believable. The challenge with Mica was to achieve this quality level while also fitting within the performance envelope of the hardware.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 THE MICA CHARACTER

Mica is based on a composite of different individuals, and artistic design unique to her. The base 3D model and textures were acquired by scanning an actor in a high resolution photogrammetry system. The models were then artistically sculpted into final form. The real-time character asset that runs on Magic Leap One is derived from a level-of-detail chain anchored by a high-quality, "VFX style" asset that can be rendered offline in a path tracer. We found that it was important for quality control and reference purposes to always have a target look for the character at this level. When making changes, we usually go back to the top-most LOD and let the changes migrate down, rather than changing directly in a lower LOD.

We use Facial Action Coding System (FACS) based blendshapes in our facial animation system with over 800 shapes for Mica's facial animation. The deformation of the eyes, tongue, teeth, jaw, head, and neck are driven by blendshapes and linear blend skinning. This skinning of the body and clothing is computed by manually creating example poses and a set of joints, and solving for the joint positions and skin weights with convex quadratic programming [Jacobson et al. 2014].

The body deformations follow a layered scheme with a base skeleton driving a higher level system of joints, and those joint poses driving blendshape and skinning based deformations. Cloth simulations are run with a range of motion animation sequence, and the resulting deformations are decomposed with the same system described above.

3 ACHIEVING REAL-TIME PERFORMANCE

When we began developing Mica, the hardware specifications for the Magic Leap One were not yet finalized. This meant that we needed to be flexible with our software stack. Unreal Engine 4 provided us with the desired combination of high performance, and

an open architecture that would permit modifications. The Magic Leap SDK is also integrated into the engine, which allows access to features like head-pose, hand-tracking, and the 6-DOF controller.

Our character rig is built in Maya, and also needs to work in engine. We knew that the asset would be modified many times during the course of production, and we needed to maintain one-to-one parity between the two systems. The rig is constructed using a custom rule based system in Maya that captures all of the procedural relationships, such as set driven keys. The rig logic is then re-established when the character asset is imported into Unreal. We extended the Unreal animation system with custom Blueprint operators for radial basis function (RBF) based data interpolation, and for running Tensorflow-Lite neural networks.

The dense hair groom is converted to polygon strips for performance reasons. Mica's hair is not dynamic, however the hair strips react to facial expressions and head motion through synchronized blendshapes. We use a custom hair shader which provides more artistic controls than the default UE4 hair shader.

3.1 Helper Joints

Secondary deformations of the body are achieved through a system of joints parented to primary skeletal joints. For example, in the left calf system, the knee and foot joints together drive a single joint positioned mid-calf, which controls the calf deformation as the leg moves. These joints, called helper joints, are placed manually in Maya when the character rig is designed.

The complex, typically non-linear, relationships between the driver joints and the helper joints have traditionally been captured using sparse data interpolation techniques employing radial basis functions. We perform this regression using multilayer feedforward neural networks with significantly fewer parameters. We found that a fully connected network with a single hidden layer, using ReLU activation functions, could achieve the same results as a much larger RBF based system. We trained separate networks for each component of motion (rotation, translation, scale) for each joint system. Translation values are scaled to 0 to 1 range for the regression network. This system runs on the CPU in 0.9ms per frame.

Our system represents rotations with quaternions. Existing methods for computing quaternions in neural networks measure loss with Euclidean angles [Pavllo et al. 2019], or retain orientation ambiguities [Mahendran et al. 2017]. Thus regressing quaternions required a new loss function combining mean squared error loss of the component values with a penalty for component values greater than 1. Since we only have unit quaternions, no component should have a value higher than 1. The training also clamps the magnitude of the gradients used for back-propagation.

The training data for the helper joint networks consists mainly of a range-of-motion animation sequence, and other example sequences. The number of hidden dimensions to use is best discovered by hyper-parameter tuning, though we found good results using the heuristic: $10 \times max(\#\text{Drivers}, \#\text{Helpers})$.

3.2 Rendering

UE4 offers three rendering paths: a mobile-optimized forward shading renderer, a desktop deferred renderer, and a version of the desktop renderer without a G-buffer that is informally referred to as "forward+". We tested all three rendering paths and found forward+ to be the best match for our needs. Some of the advantages of forward+ over the mobile forward renderer include a wider choice of lighting and surface shading models, compute-shader optimized skinning and post-morph-target normal recalculations, and more post-processing options. Access to engine source code has been vital, as we routinely make custom changes to character-related engine features and shading models.

Magic Leap's hardware is capable of running in full deferred mode but maintaining 60 frames per second in stereo leaves little room for other runtime computation. Performance becomes primarily G-buffer bound, leaving unused cycles on the GPU. Forward+ frees us from that bottleneck at the cost of some engine features due to the lack of a gbuffer.

Our hardware and software stack supports Vulkan and we worked closely with Epic as they rolled out support for it in the engine. Vulkan freed up room on the CPU and allowed us to shuffle our allocation of CPU vs. GPU resources to get us over the 60fps stereo threshold.

4 BEHAVIORAL SYSTEM AND EXPERIENCE DESIGN

One of the most compelling things about Mica is her ability to hold the viewer in her gaze. ML-1's spatial computing framework provides a consistent coordinate frame based on the real-world environment. The character and the user exist in this frame, and the guaranteed low-latency head tracking allows her to look at the viewer's eyes directly and accurately. The Perception framework also continuously scans the real world space for geometry, so Mica stands where the floor actually is. Her body can also move based on her orientation with the viewer. Mica's gaze and body disposition are also dynamically determined by a discomfort system that avoids holding poses for unnaturally long periods of time.

ACKNOWLEDGMENTS

Mica could not have been created without the significant contributions of: James Bancroft, Nafees Bin Zafar, Victoria Bishop, Sean Comer, Fabian Elmers, Prudence Fenton, Chris Hebert, Takashi Kuribayashi, Kamy Leach, Victor Leung, Jeff Lin, Jonathan Litt, Thomas Miller, John Monos, Mark Orbik, Rebecca Paris, Dan Platt, The Hung Quach, Scot Shinderman, Geoff Wedig, Aaron Wilson, Alice Wroe, and Joelle Zimmermann.

REFERENCES

Epic Games. 2019. Unreal Engine 4 Documentation: Digital Humans. Retrieved February 9, 2019 from https://docs.unrealengine.com/en-us/Resources/Showcases/DigitalHumans

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and JP Lewis. 2014. Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH 2014 Courses*.

S. Mahendran, H. Ali, and R. Vidal. 2017. 3D Pose Regression Using Convolutional Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 494–495. https://doi.org/10.1109/CVPRW.2017.73

Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. 2019. Modeling Human Motion with Quaternion-based Neural Networks. *CoRR* abs/1901.07677 (2019). arXiv:1901.07677 http://arxiv.org/abs/1901.07677

Mike Seymour. 2018. Visual Disruptors podcast : Ep 2 Digital DOUG. Retrieved February 9, 2019 from https://www.fxguide.com/quicktakes/visual-disruptors-podcast-ep-2-digital-doug/