Dynamic Mixed-Reality Compositing with Unity

Joanna Tarko[1, 2]
j.k.tarko@bath.ac.uk
Christian Richardt[1]
c.richardt@bath.ac.uk
Peter Hall[1]
p.m.hall@bath.ac.uk

[1] Department of Computer Science, University of Bath
[2] Checkmate VR Ltd.

We present a system for **dynamic mixed-reality compositing**, or how to insert *dynamic* computer-generated (CG) elements into live-action video footage in real time. The goal of compositing is to combine visual content from different sources, such as live-action footage, still images and animations, in a way that they match each other regarding colour, lighting, scale, perspective, camera movement and timing. Most of these aspects can be matched using geometric calibration of the camera and mixed-reality rendering techniques. To ensure that both sources of visual content are composited seamlessly, our approach combines the accuracy of offline camera tracking with real-time mixed-reality rendering performed in the Unity game engine.

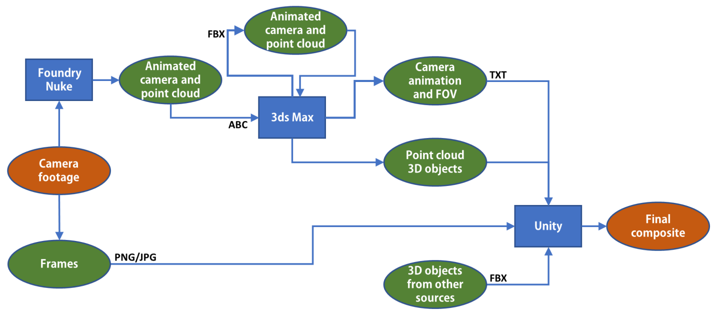

Figure 1: Dynamic compositing pipeline with Foundry Nuke for camera tracking and sparse scene reconstruction, 3ds Max for exporting camera animation and creating 3D objects, and Unity for combining all elements to produce the final composite.

Our main contribution is a practical solution for synchronising clock-driven and event-driven elements in the Unity game engine environment, to make correct graphics element insertions into live-action footage possible. We present a complete pipeline for dynamic mixed-reality compositing (Figure 1) that provides users with tools for interacting with inserted elements, manipulating them in real time, and creating any number of different versions of the same scene in a dynamic way. Our pipeline is completed by differential rendering for image-based shadowing of inserted objects.

The demands of real-time rendering mean that the graphics engine generates rendering events at random intervals within a fixed upper time bound, yet video frames in the real footage are spaced equally in time. This results in the problem of synchronising event-driven and clock-driven elements, which has to date remained unsolved and prevented graphic elements from being composited correctly into video footage. This problem does not exist in mixed-reality applications that track the camera in real time, as the virtual camera parameters are adjusted on a per-frame basis.

To overcome the synchronisation problem, we need to reject Unity tools for handling media assets and turn to a more basic approach. First, we split both camera pose animation and camera footage into their respective samples before importing them into Unity, to enable fast random access to specific samples. In the case of animation, these samples are the camera poses associated with every video frame along with the camera's field of view, which we save to a plain text file using a custom script. The camera footage, in turn, is also split into separate video frames. Second, we exploit that physics simulations in computer games require updating at constant time intervals (up to the current game refresh rate) and therefore have a dedicated update function. We set this time interval to match the video frame rate (e.g. to 1/25 s to match the frame rate of 25 fps). Every time the update function is called, our script reads the current camera position and rotation from the text file and applies it to the virtual camera. At the same time, the corresponding video frame is loaded from the assets folder and displayed as a dynamic texture on the plane aligned and linked to the camera.

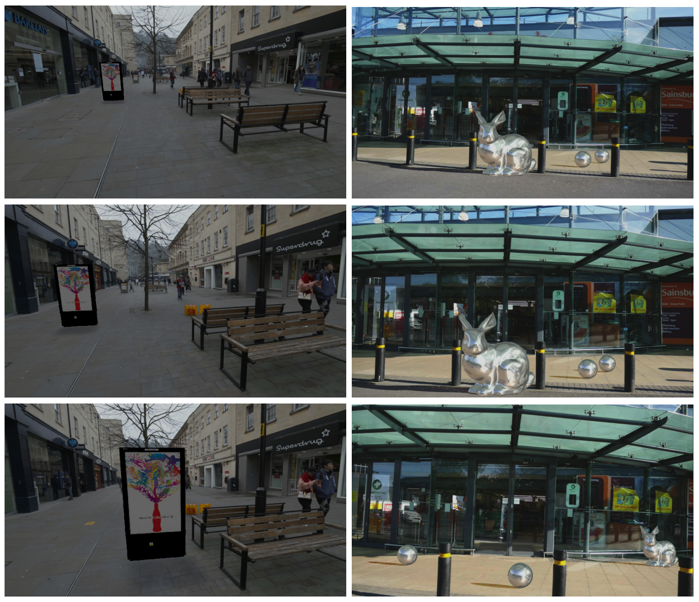

Figure 2: Scenes used to test the accuracy of synchronisation and compositing. Inserted CG elements include a poster stand (left) and a reflective bunny and two balls (right). Single frames extracted from the final composite sequence show that CG elements stay fixed in their places in the scene when the camera moves (first two rows). The video playback can be paused at any frame, and the inserted CG elements can be moved or changed using keyboard shortcuts (bottom row).

Debevec's concept of differential rendering for light transport in mixed-reality scenes [1] allows the casting of shadows onto the estimate of a local scene, such as a table top or ground plane, to be blended into the background image. In most cases, the local scene can be reduced to a simple plane. We fit this plane automatically to a user-selected subset of the point cloud recovered from sparse scene reconstruction. Instead of estimating the reflectance model of the local scene iteratively, we apply a screen-space shader to the used plane, using the footage from the camera as a dynamic texture to provide the diffuse colour, as proposed by Rhee et al. [2]. This ensures the correct colour of the produced shadows.

We tested our dynamic mixed-reality compositing pipeline on a set of videos, as shown in Figure 2. Firstly, we assess the accuracy of temporal synchronisation by observing if the inserted objects stay in the same place throughout the shot, assuming that there were no errors in the input camera tracking. Secondly, we test the interaction tools. The scenes include an implementation of differential rendering for image-based shadowing, and the second scene additionally uses an HDR map of the environment for image-based lighting. The user manually sets up light source position, colour and intensity to obtain virtual shadows that match those cast by the real objects. Both test scenes allow the user to control the video playback as well as the position and rotation of the CG objects using keyboard shortcuts.

This research was funded by the Centre for Digital Entertainment (EPSRC CDT EP/L016540/1), and by Checkmate VR Ltd. as part of their R&D programme.

[1] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH*, pages 189–198, 1998.

[2] Taehyun Rhee, Lohit Petikam, Benjamin Allen, and Andrew Chalmers. MR360: Mixed reality rendering for 360° panoramic videos. *TVCG*, 23(4):1379–1388, 2017.