Real-time 3D Character Integration into a Real-World Environment Using Reconstructed Z Depth

Tom Hart
CAPCOM CO, LTD.
Osaka, Japan
thomas-hart@capcom.com

Minoru Nakai
CAPCOM CO, LTD.
Osaka, Japan
minoru-nakai@capcom.com

Figure 1: *Shooting video on the left; the same shot with reconstructed z depth and 3D real-time elements on the right.*

1. Introduction

We present a novel and efficient pipeline for interactive entertainment integrating a real-time 3D character into real-world environments. Instead of conventional 3D environment creation, our method reconstructs the z depth of real-world video footage, allowing a 3D character to navigate it in real-time. Three processes afford this: 3D match moving, photogrammetric 3D model generation of environments, and HDRI lighting. 3D match moving software determines the position of each real-world camera shot in relation to 3D model space. Environment geometry is rendered as invisible shadow-catchers, beyond which video footage itself is projected as a texture. 3D characters, for example, are then added, lit with HDR images of the location, and made interactive in a game engine.

2. Overview

The field of interactive entertainment has gifted us control of real-time characters in photorealistic environments from Sega's Astron Belt in 1982 to the handcrafted world of Lumino City in 2014. However, we imagine bringing users the freedom to explore as a real-time 3D character, real-world locations precisely as captured in HD video. The most fundamental challenge to realise this goal is reconstructing the z depth of 2D video footage, with accurate perspective and lighting to integrate real-time elements such as the user's avatar. We propose a new approach utilising a reference object to unify scale and position across multiple software, almost entirely automating the reconstruction of interactive video footage z depth. Our core innovation is the efficient matching of position, rotation and scale of each tracked real-world camera shot into a single scanned 3D environment model, resulting in real-world video footage which is traversable by and correctly occludes real-time characters as well as other elements.

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792518

3. Related Work

Interactive entertainment integrating real-time 2D characters with real-world video date from 1982 to present day. Instead, our approach focuses on real-time 3D characters in real-world space. Real-time 3D characters in pre-rendered environments appeared in the 1990s (Capcom's own Resident Evil a good example in 1996). However, their creation was a time-consuming process of handcrafting environments in 3D, then using proprietary tools and hand-placed geometry to integrate the real-time elements. Our approach automates the majority of collision detection creation and lighting, giving interactive real-world locations in minimal time. Other related work includes advancements in Augmented Reality (AR) which promise to integrate real-time 3D characters into a user's location. However, lighting 3D characters and the occluding of them by surrounding objects are current hurdles in AR which our method overcomes, and unlike AR our method allows users to be transported to a real-world location other than where they are currently, a feature clearly distinguishing the two.

4. Implementation

To test our approach, we constructed a prototype scene with 20+ camera cuts of a single real-world space over 100m long. We took free move video shots of the location (i.e., those with parallax) from various angles to later allow full navigation of the area, incorporating a static reference object for match moving. Concurrently we took a few hundred photos including the same reference object (used to generate geometry of the environment in Photoscan), and a 360 degree HDR photo per video shot (used to light additional 3D real-time elements). Off location, we tracked the reference object with match moving software Syntheyes, recreating a 3D version of the reference object in correct spatial relation to each camera. Then in Blender we aligned this to the position, rotation and scale of the reference object generated as part of the Photoscan 3D environment. By aligning the two 3D reference objects in the same 3D space, all other elements (camera and collision data) are aligned automatically. Taking this data into Unity, we used invisible shadow-catcher shaders on the geometry, and HDRI based Lighting via Marmoset Skyshop. Real-time 3D objects are then able to navigate this invisible shadow-catching 3D environment, onto which video footage is projected, becoming the photorealistic colour and light information.