LAYERED VIEW-DEPENDENT TEXTURE MAPS

Marco Volino[1], Julien Pansiot[2], Oliver Grau[2] and Adrian Hilton[1]

[1]University of Surrey, UK, {m.volino, a.hilton}@surrey.ac.uk
[2]BBC R&D, UK, {julien.pansiot, oliver.grau}@bbc.co.uk

Abstract

We present a novel layered texture map representation for use with multi-view 3D video sequences. The texture information for each frame is stored in multiple texture maps which maintain view-dependence during the rendering process. An evaluation of the representation demonstrates that it reduces the storage and memory requirements with no significant reduction in rendering quality when compared to a state-of-the-art free-viewpoint video renderer.

Keywords: View-Dependent Rendering, Texture Mapping

1 Introduction

In this paper, we assume that the 3D mesh of an actor has been previously generated using video captured from multiple cameras. The use of multi-view capture to produce 3D content has become increasingly popular. One drawback is that even a short multi-view sequence requires large amounts of storage. Previous work on multi-view texture maps has sought to create a single texture map by combining the information to reduce seams between camera views. This effectively throws away the view-dependent information which is important to maintain the realism of the captured video. Free-viewpoint rendering achieves photo-realistic results by sampling directly from the originally captured frames based upon the desired viewpoint. Whilst this method enhances the rendering quality, it also dramatically increases the amount of data required. We propose a novel method to store the original camera texture information into layers, which maintain view-dependent information while reducing the storage requirements.

2 Layered Texture Map Representation

The layered texture maps are created in an offline process and require spatio-temporally aligned geometry. Texture coordinates are generated for the first frame. These are valid for every frame in the sequence and are used during rendering. The extraction process ranks the view of each face of the mesh using the angle between the face normal and the camera direction. Views with the same ranking are compiled into a single texture map. A camera assignment map is also generated showing the camera each view originated from. The camera assignment map and texture map pair (Figure 1) form a layer. The maximum number of layers which can be generated is equal to the number of cameras used to capture the scene. This extraction is performed for every frame in the sequence. The rendering algorithm reproduces the result of the free-viewpoint video renderer (FVVR) [1] by searching through the layers to find the same visual information that would be used in the FVVR algorithm. When the same information is not available, the layered free-viewpoint video renderer (LFVVR) uses all available texture information to generate the final texture.

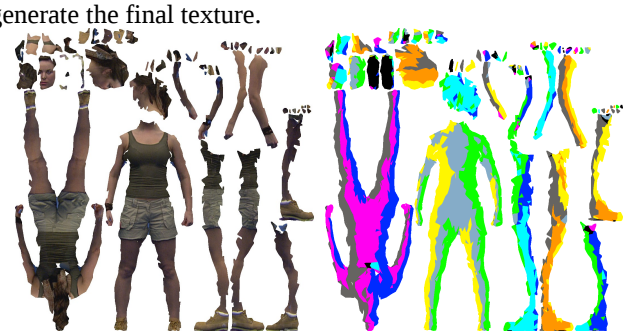

Figure 1: Texture map (left), Camera assignment map (right)

3 Results

The LFVVR was compared to an open source, state-of-the-art FVVR [1] in terms of storage, memory and rendering quality. The test sequence consisted of 31 frames captured from 8 views. Figure 2 shows that as the number of layers (L) is increased, the quality increases up until L=4 after which no significant increase occurs. This is the case for all number of texturing cameras (M). This occurs at L=4 due to the set-up of the capture studio which uses eight cameras positioned uniformly around a circle meaning the majority of mesh faces have four or less views.

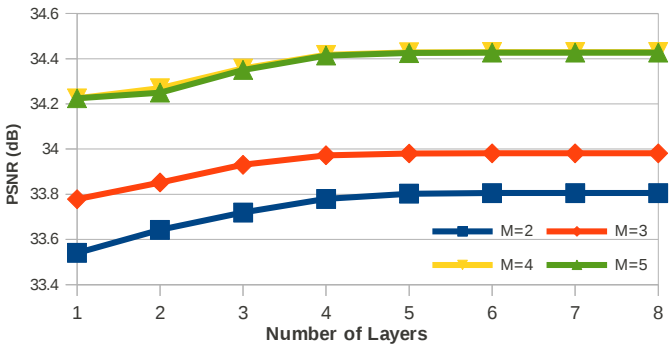

Figure 2: Rendering quality of LFVVR against FVVR

Using four layers reduces the required storage per frame by 84% and by 95% when encoded into a video compared to the captured frames. The GPU memory required is also reduced by 44% which is directly related to the texture map size.

References

[1] J. Starck, J. Kilner & A. Hilton (2009): A Free-Viewpoint Video Renderer, Journal of Graphics, GPU, and Game Tools, 14:3, 57-72