LOW COMPLEXITY CONNECTIVITY DRIVEN DYNAMIC GEOMETRY COMPRESSION FOR 3D TELE-IMMERSION

Rufael Mekuria, Pablo Cesar, Dick Bulterman

Centrum Wiskunde & Informatica

ABSTRACT

Geometry based 3D Tele-Immersion is a novel emerging media application that involves on the fly reconstructed 3D mesh geometry. To enable real-time communication of such live reconstructed mesh geometry over a bandwidth limited link, fast dynamic geometry compression is needed. However, most tools and methods have been developed for compressing synthetically generated graphics content. These methods achieve good compression rates by exploiting topological and geometric properties that typically do not hold for reconstructed mesh geometry. The live reconstructed dynamic geometry is causal and often non-manifold, open, non-oriented and time-inconsistent. Based on our experience developing a prototype for 3D Tele-immersion based on live reconstructed geometry, we discuss currently available tools. We then present our approach for dynamic compression that better exploits the fact that the 3D geometry is reconstructed and achieve a state of art rate-distortion under stringent real-time constraints.

Index Terms— 3D Tele-immersion, dynamic geometry compression, 3D Mesh Compression, tele-presence

1. INTRODUCTION

Advances in 3D reconstruction and rendering – and the success of inexpensive consumer grade depth cameras – enable the creation of highly realistic representations of the participants as triangle mesh models. For efficient interactive transmission of such streams over bandwidth limited link (dynamic) geometry compression becomes necessary. Figure 1 shows the envisioned stages in 3D tele-immersion with live reconstructed geometry. The participant (a break dancer in this case) is captured from multiple angles with consumer grade depth cameras. From these depth images a 3D mesh is created that is subsequently compressed with dynamic geometry compression and transmitted over the IP network and rendered remotely in a 3D space. The 3D mesh representation offers flexibility in rendering and shading based on stereo or with multiple views. Also, it allows integration in virtual worlds where this representation is common.

![Figure 1: Live 3D Reconstruction with depth cameras followed by dynamic geometry compression, IP transmission and remote composited rendering (3D Tele-immersive pipeline)](image)

Ported as a full 3D Mesh, the participant is truly immersed into a virtual world. An example of such a full system based on mesh is described in [1] and based on point cloud meshed at the receiver in [2]. This paper proposes a connectivity driven mesh codec optimized for 3D Tele-immersive mesh data. We present related work and tools in section 2 and 3 respectively and propose our method in section 4. We evaluate the approach with a publicly available dataset used for 3D Tele-immersive Mesh coding in Motion Picture Experts Group (MPEG) in section 5.

2. RELATED WORK

Fast frame compression has been a bottleneck in 3D Tele-immersive systems design, and several methods have been proposed such as [3] and [4] that support real-time compression of multiple views and depth for 3D Tele-immersion (but not 3D Mesh data). Some recent methods to support 3D Tele-immersive time varying geometric data include [6] based skinning and motion prediction based on the skeleton motion and [7] based on exploiting the grid pattern when 3D data is reconstructed via a grid-pattern based scanning system. Nevertheless, a large body of research on static and dynamic mesh compression research is still available (see [8] for a survey) that have not been tested in 3D Tele-immersive applications. An example of a dynamic mesh codec for time-consistent geometry is described in [9], which is part of the ISO MPEG-4 standard. Also, 3 different methods for time-varying mesh geometry compression have been briefly described in [10]. In 3D Tele-immersion, geometry is often accompanied by attributes for normals, colors and textures that are often not handled by all existing mesh codecs. Also, 3D Tele-immersive 3D Meshes can be time-inconsistent (varying connectivity per frame), non-manifold, noisy, non-oriented and can consist of multiple isolated components. A recent research on compression of CAD data with non-manifold and isolated components revealed that some state of art methods handle this type of data inefficiently [11]. Instead, in this paper we propose a method for time-inconsistent geometry compression and compare it to existing tools available in MPEG-4 [12] suitable for 3D Tele-immersion.

3. AVAILABLE TOOLS

Recently Motion Picture Experts Group (MPEG) has standardized mesh codecs that are applicable to triangular meshes with any geometric property in MPEG-4 SC3DMC. MPEG TFAN [13] achieves compression by composing the connectivity in fans of triangles. These fans can then be efficiently coded by assigning less bits to the most common fan configurations. This achieves good compression performance. Unfortunately, the composition into triangle fans is computationally slightly expensive, as it...
We consider the 3D mesh sequences reconstructed on-the-fly, as a geometry and attribute data. The idea behind the connectivity coding approach presented in this paper is that 3D reconstruction can introduce specific regularities in the connectivity information that can be exploited for compression purposes. For example, in the zipper method in [17] multiple range images are tessellated first into range surfaces that are subsequently zippered (stitched) together (after redundant triangles are removed). If these range surfaces are tessellated in consistent order, this can introduce a more regular and predictable connectivity structure. Such patterns were also found in the connectivity of the reconstruction data in [5]. As such, patterns occur many times in a row, we actively search for them and use them for efficient encoding. An example of how following differences occur is shown in Table 1, where each next index is an increment of 1. While such patterns might differ per reconstruction method and need to be found before they can be exploited, it is very beneficial to enable low-complexity encoding of large meshes.

<table>
<thead>
<tr>
<th>Pattern Type</th>
<th>Diff1</th>
<th>Diff2</th>
<th>Start</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern 1</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>1</td>
</tr>
<tr>
<td>Pattern 2</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>1</td>
</tr>
<tr>
<td>Pattern 3</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1 Patterns of connectivity indices with repetitive difference in the connectivity list

Figure 3 illustrates the connectivity compression scheme. First, the entire connectivity information is searched for repeated regularities which are counted and stored in the data-structure pattern run shown in Table 2. The mode field represents the type of pattern, the diff fields the two differences that occur and the count field signals the number of repetitions. The start field is used to reference the position of the first index in the connectivity list. This value is used to detect the start of a run in the next encoding step and in the decoder. Next, all indices are iterated again. If an index is the start of a run the pattern run (indicated by the start field), is stored and the connectivity index iterator is increased with the count field. If an index is not stored in a run, the difference with the index in the previous face is stored instead. Storing differences instead of absolute values yields mostly small numbers skewed around 0 and 1 and therefore allows efficient entropy coding. The resulting data vector is entropy encoded via the zlib library [18].
5. EXPERIMENTAL RESULTS

5.1. Datasets

For evaluation of the compression method we use datasets of meshes reconstructed with five depth cameras that are currently used in the 3DG group of Motion Picture Experts Group (MPEG) for evaluation of 3D Tele-immersive Mesh coding technologies. They have been created by the Center for Research and Technology Hellas (CERTH) based on the method reported in [5] and contain participants performing different activities. The datasets are publicly available at the website currently hosted at [16]. These datasets represent the case where a user is in a room captured by multiple depth cameras at a distance of 300 cm. We have also performed several test sets with data reconstructed with one depth-camera representing a user that is sitting in front of his computer (at around 1 meter).

5.2. Quality Evaluation Metrics

We utilize two metrics to assess the geometric quality of the mesh the symmetrical Hausdorff distance and similarly the symmetrical root mean squared distance, computed between the original and decoded surface. We use the symmetrical Hausdorff distance which is defined in equation (6) as:

\[ d_{sym}(M,M') = \max\{d(M,M'), d(M',M)\} \] (1)

Where the M is the original and M' the decoded mesh and d(M,M') is the Hausdorff distance between the two surfaces. Similarly the symmetrical root mean square error (RMS) defined where instead d(M,M') is the root mean square distance between the two surfaces. To facilitate the computation of these metrics we use the tool developed in [19] (see also for additional details on the used metrics). Alternatively, we compare the quality of the colors and normal (appearance) and the normal with root mean square error on a per vertex basis.

\[ d_{app} = \frac{1}{\sqrt{|M|}} \sum_{p \in M, p' \in M'} ||p - p'||_2 \] (2)

Where p and p' denote the 3 coordinate color/normal in original mesh M and decoded mesh M' respectively. As the tfa codec does not preserve the order of the vertices we compare to the SVA codec at the same quantization parameter. Also, we compare against the quantization error introduced by quantization of a uniform source with a uniform quantizer in 3D which is given by

\[ d_{uniform,quantization,uniform,source,rms} = \sqrt{3\Delta^2} \] (3)

Where \(\Delta\) is the quantization step size. Lastly we compared the encoding and decoding times in the proposed codec and provide screenshots of the original and decoded meshes.

5.3. Comparative Results

Our scheme was implemented in C++ and the test machine was a desktop machine with Intel i7 CPU and 8.00 GB or RAM and a GeForce GTX 760 video card. The MPEG SC3DMC Codec were compiled from source code with the same compiler as available on...
To avoid overhead that deals with loading the supported text formats in MPEG part-25 [21], we interfaced directly the class SC3DMEncoder and SC3DMCDecoder in the reference software with an MPEG indexed face set structure directly loaded from the input mesh. All files are first loaded into memory before the compression routine is started. The run times are recorded with CPU wall clock times provided by the boost C++ library with a resolution around 366 ns.

We ran all methods on N=158 Meshes with 5 depth cameras with average of 302K Vertices. Figure 4 shows a large speedup is achieved compared to other methods. In terms of compression size, we achieved on average a mesh size of 1,460 KiloBytes compared to 1266 KiloBytes with TFAN MPEG with 10 bits QP and 6 bits colors and normals. This is a 15% larger file size compared to TFAN but achieves a speedup of almost 10 times. The result is similar in case the meshes are reconstructed with 1 depth camera (average of 72K vertices), but here we encode meshes in as little as 35 ms on average.

Evaluation of the symmetric quality metrics shows that comparable quality of geometry is achieved (Figure 5). The jump in the right side of the graphs in Figure 5 represents the fact that more densely sampled meshes are tested there (more vertices and visual detail). The results with our method are still closely clustered around 0.0002 and the quality is approximately equal to 10 bits quantized mpeg compression (a bit worse with sparser meshes and a bit better with denser meshes). The method achieves higher quality decoded models when compared to 8 bits quantization.

Figure 5 Encoded and decoding results with 5 cameras

Evaluation of the symmetric quality metrics shows that comparable quality of geometry is achieved (Figure 5). The jump in the right side of the graphs in Figure 5 represents the fact that more densely sampled meshes are tested there (more vertices and visual detail). The results with our method are still closely clustered around 0.0002 and the quality is approximately equal to 10 bits quantized mpeg compression (a bit worse with sparser meshes and a bit better with denser meshes). The method achieves higher quality decoded models when compared to 8 bits quantization.

The results show that our method achieves slightly lower distortion of the normal data and the colors compared to the theoretical and the measured values. In Figure 7 shows snapshots of the decoded meshes, most artifacts are related to the 3D reconstruction method and not the compression method.

6. CONCLUSIONS

This paper introduced a connectivity driven method for dynamic geometry compression of 3D Tele-Immersive Mesh Sequences. The method for connectivity coding achieves highly reduced computational complexity compared to MPEG TFAN and is much more efficient than MPEG SVA. The geometry compression with delayed differential quantization reduces quantization artifacts introduced by low quantization parameters. With mostly 4 bits differential quantization a comparable quality to 10 bits uniform quantization was achieved. As entropy coding of the geometry information is avoided this resulted in much lower computational latency. The method for connectivity coding is currently under evaluation in a core experiment for an extension of the MPEG-4 SC3DMC standard.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. ICT-2011-7-287723 (REVERIE project).
References


[16] M. Levoy, G. Turk, and, "Zippered polygon meshes from range images," in 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH '94), 1994, pp. 311-