2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)

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.

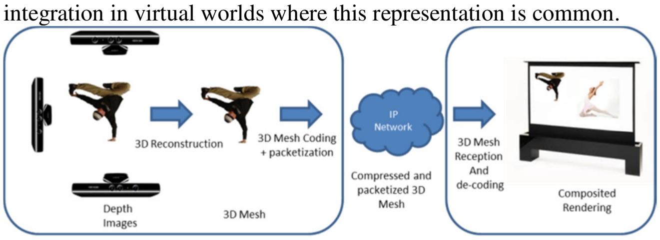

Figure 1 Live 3D Reconstruction with depth cameras followed by dynamic geometry compression, IP transmission and remote composited rendering (3D Tele-immersive pipeline)

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 requires finding the adjacency relationship between vertices and triangles. Therefore, MPEG standardized the simpler SVA codec that is better suited to low-delay compression of reconstructed mesh data. This method exploits repeating vertices in the triangle list ("shared vertices"). In our approach we extend this concept by looking at patterns that occur in the mesh connectivity over a larger number of vertices, obtaining an increase in compression rate over MPEG SVA [14]. Another tool available for handling geometry is the point cloud library [15], while it provides an octree coder for point clouds, no full mesh codec is provided. Instead we adopt the connectivity driven approach as in TFAN or SVA, coding the connectivity first, followed by differential prediction of the geometry and attribute data.

4. LOW COMPLEXITY CONNECTIVITY DRIVEN MESH CODEC

We consider the 3D mesh sequences reconstructed on-the-fly, as a temporally incoherent mesh sequence MS. That is MS = $M^i = (V^i, F^i), i = 0...n$ is a reconstructed mesh sequence and the number of vertices in each mesh $|V^i|$ and faces $|F^i|$ is not constant over i. We aim to compress this data with a rate-distortion comparable to available methods, but with a lower computational complexity resulting in real-time encoding. We evaluate the rate-distortion-complexity with metrics defined in Section 5 and the dataset in [16].

Figure 2 outlines the compression system. We introduce a fast connectivity traversal method combined with connectivity-driven (offline) optimized DPCM coding and layered quantization. We explain the rationale and operations for compressing the connectivity $C(E)$ in the next sub-section. The differential encoding of the geometry $G(V)$ followed by non-linear quantization is presented in 4.2. In section 4.3. we discuss how the appearance (normals(V) and colors(V)) are handled (appearance quantization).

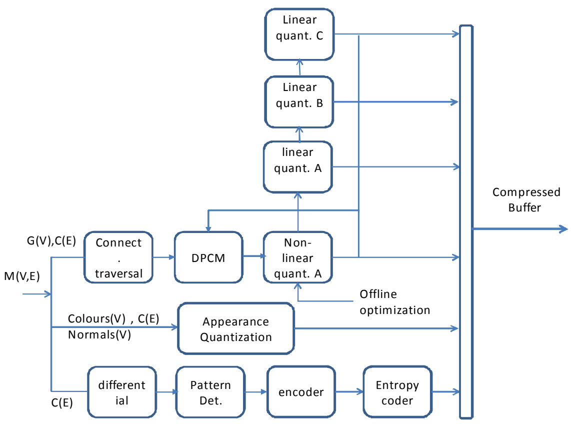

Figure 2 Schematics of proposed dynamic mesh codec

4.1. Pattern Based Connectivity Coding

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 zippering 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 a 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.

1632 1520 1633
1520 1521 1633
1633 1521 1634
1521 1522 1634

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].

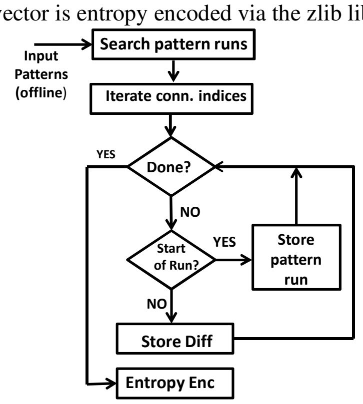

Figure 3 Pattern Based Connectivity Coding

Mode	Diff1	Diff2	Start	Count

Table 2 Structure Pattern Run

4.2. Geometry coding with delayed differential encoding

Uniform quantization is used for geometry compression in MPEG-4 SC3DMC and other methods, which have been designed with high quality graphical models with large quantization parameter (QP 10-20). For the low-bit rate requirements for streaming live reconstructed 3D geometry that are densely sampled this method however poses several disadvantages.

Firstly, the dynamic range of the vertices that are quantized in a 3D space is large compared to the details in the 3D surface that becomes visible to the viewer. This makes details in the surface vulnerable to quantization distortion that increases with low-quantization parameters. When low quantization parameter are used for geometry compression, the lack of detail in the mesh surface immediately becomes visible and vertices may share positions resulting in degenerate triangles (faces with a zero surface) and blocky appearance of the surface arises.

Therefore, instead we propose quantization after the transform (DPCM) with a variable number of bits. In our case we propose a layered structure: most vertices are quantized with 4 bits, but values outside the 4-bits range are quantized with 8 bits, 16-bits and 32-bits respectively when needed, i.e., if the differences become large. This way we avoid large errors in the geometry, as even for a small amount of vertices large quantization errors can become clearly visible when the mesh is rendered.

We code differences between vertices that are connected to each other. In this case the differences are even more fine-grained as by the nature of densely sampled 3D reconstruction, connected vertices are closely co-located. The primary quantization vector codes over 95% of the values and is defined as for the given datasets:

$$\begin{bmatrix} -0.011 - 0.0063 - 0.0042 - 0.00315, -0.00236 \\ -0.0012 - 0.0004\ 0\ \ 0.0004\ 0.0012\ 0.00236 \\ 0.00315\ \ 0.0042\ 0.0063\ 0.011 \end{bmatrix}$$

In this configuration for the 4 bits non-linear quantization vector was computed offline. It can be dependent on the reconstruction method, we envision that such information can be exchanged in a 3D Tele-immersion system out-of band (in our 3D Tele-immersion system we use a custom session management system that can exchange such information). Subsequently, we utilize a layered structure with 8-bits for the range from 0.11 to 0.1, and 16bit between 0 and 2 to cover the entire dynamic range (in this case we use the value of the primary quantization vector for sign indication). This information again depends on the reconstruction system and can also be send out of band when a user joins a 2D-Tele-immersive session with a specific setup.

To perform connectivity driven differential encoding, we defined and implemented a traversal method that does not changes the order of the vertices in the list. Each of the faces is traversed, and the first connected vertex that was previously stored (set) is used for differential prediction. If none of the connected vertices have been previously traversed and set, we set the first vertex index in the face by adding this value to the reserve_vals list. The other connected vertices in the face are then differentially encoded. At the decoder, we traverse the connectivity in a similar manner, unset vertices are loaded from the reserve_vals list and the other vertices are recovered via inverse differential prediction. This method works well as the reconstructed meshes are relatively coherent (connected vertices are also closely located in the indexed face list). In practice less than 0.1% of the vertices are stored in the reserve list in some of the reconstruction systems we have tested. Values in the range $[-0.011\ ,\ 0.011]$ are stored and appended in the 4-bits vector, values between $[-0.1\ -\ 0.1]$ are quantized with 8 bits and appended to the eight bits queue vector. Larger differential values are stored in the 16-bits or 32 bits queue but occur very rarely but need to be stored accurately as they introduce large visual distortion when decoded incorrectly.

4.3. Appearance Quantization

In the case of surface mesh geometry, the perceived appearance depends not only on the color but also on the geometry coordinates and surface normal data. We deploy the same system of late differential quantization for the appearance data (normal and colors), however the layers are assigned differently. We quantize normal components in the range from -0.25 – 0.25 with 4 bits and the rest with eight bits (8 bits plus the sign bit retrieved from the 4 bits quantization vector). We code color differences between -25 and 25 with 4 bits and the rest with 8 bits (precisely 8bits plus the sign bit retrieved from the 4 bits quantization value covering the entire range$[-255,255]$. Again, such information can be communicated out of band prior to the media streaming session to configure the encoder and decoder properly.

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 Haussdorf distance* and similarly the *symmetrical root mean squared distance,* computed between the original and decoded surface. We use the symmetrical Haussdorf distance which is defined in equation (6) as:

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

Where the M is the original and M' the decoded mesh and d(M,M') is the Haussdorf 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.

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

Where p and p' denote the 3 coordinate color/normal in original mesh M and decoded mesh M' respectively. As the tfan 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

$$d_{uniform\_quantization\_uniform\_source\_rms} = \sqrt{3}\Delta/\sqrt{12} \qquad (3)$$

Where Δ 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 Codecs were compiled from source code with the same compiler as available on

[20]. To avoid overhead that deals with loading the supported text formats in MPEG part-25 [21], we interfaced directly the class SC3DMCEncoder 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% lager 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.

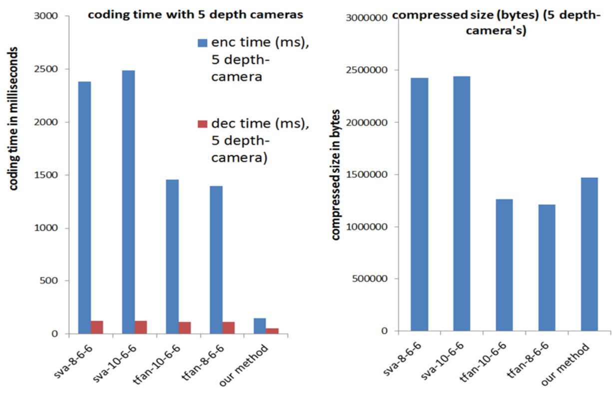

Figure 4 Encoding and decoding results with 5 cameras

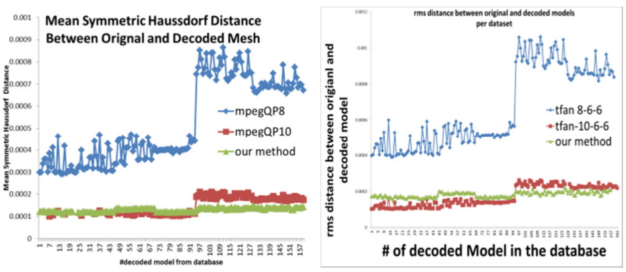

Figure 5 Mean symmetric Haussdorf distance between the original and decoded model

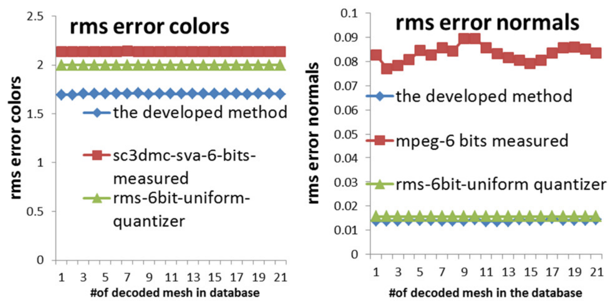

Figure 6 RMS error of color and normal between original and decoded reconstructed meshes

Figure 7 Decoded meshes, the original (left), mpeg (tfan) with QP10 (middle), our method (right)

We compared the quality of the color and normal data with the root mean square difference on a per vertex basis. As TFAN re-orders vertices we compared only with SVA, (Figure 6) and $\sqrt{3}\Delta/\sqrt{12}$ which is the root mean square quantization error introduced when quantizing a uniformly distributed source uniformly in 3D with step size Δ.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

[1] R.Mekuria, M.Sanna, S.Asioli, E.Izquierdo, D.C.A. Bulterman, P.S.Cesar , "3D tele-immersion system based on live captured mesh geometry," in *Proceedings of the 4th ACM Multimedia Systems Conference (MMSys '13). ACM*, Oslo, Norway, 2013, pp. 24-35.

[2] C. Zhang, Q. Cai, P. A. Chou, Z. Zhang, R. Martin-Brualla , "Viewport: A Distributed, Immersive Teleconferencing System with Infrared Dot Pattern," *IEEE Multimedia Magazine*, vol. 20, no. 1, pp. 17-27, Jan-March 2013.

[3] K.M. Patel, H.Fuchs. S.U. Kum, "Real-Time Compression for dynamic 3D Environments," in *ACM Multimedia'03*, Berkeley, CA, USA, 2003.

[4] Z. Yang, Yi Cui, Z. Anwar, R. Bocchino, N. Kiyanclar, K. Nahrstedt, R. CampBell, W. Yurcik , "Real-Time 3D Video Compression for 3D Tele-Immersive Environments," in *MMCN'06*, 2006.

[5] I. Daribo, R. Furukawa, R. Sagawa, H. Kawasaki, S. Hiura, N. Asada , "Efficient Rate-Distortion Compression of Dynamic Point Cloud for Grid-Pattern-Based 3D Scanning Systems," *3D Research Journal,* vol. 3, no. 1, 2012.

[6] A. Doumanoglou, D. Alexiadis, D. Zarpalas, P. Daras , "Towards Real-Time and Efficient Compression of Human Time-Varying-Meshes," *IEEE Transactions on Circuits and Systems for Video Technology*, Aug 2013.

[7] A.Smolic et al., "A survey on coding of static and dynamic 3D meshes," in *Three-Dimensional Television*.: Springer, 2008, pp. 239-311.

[8] K. Mamou, T. Zaharia, F. Preteux , "FAMC: The MPEG-4 standard for Animated Mesh Compression," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, San Diego, 2008, pp. 2676 - 2679.

[9] Aizawa, Kiyoharu , "Compression of 3D Video:Various Approaches to Compress Time Varying Meshes," in *IEEE International Conference on Image processing*, 2008.

[10] K. Mamou, T. Zaharia., F. Preteux , "3D Mesh Coding Techniques Applied to CAD Data: A Comparative Evaluation," in *Conference on Intelligent Multimedia Documents on Internet - MediaNet2004*, Tozeur (Tunisia), 2004.

[11] MPEG, "Information technology -- Coding of audio-visual objects -- Part 16: Animation Framework eXtension (AFX)," ISO, International Standard ISO/IEC 14496-16:2011, 2011.

[12] T. Zaharia, and F. Preteux K. Mamou, "TFAN a low complexity 3D Mesh Compression algorithm," in *Computer Animations and Virtual Worlds (CASA'09) pp. 343-354*, 2009.

[13] E.S. Jang., S. Lee,B. Koo, D. Kim,K. Son , "Fast 3D Mesh Compression using Shared Vertex Analysis," *ETRI Journal*, vol. 32, no. 1, 2010.

[14] R.B. Rusu, "3D is here: Point Cloud Library(PCL)," in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 1-4.

[15] D. Alexiadis. (2013, Oct.) Datasets of Multiple Kinects-based 3D reconstructed meshes. [Online]. "http://vcl.iti.gr/reconstructions/"

[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. 311-318.

[17] D. Alexiadis, D. Zarpalas, and P. Daras., , "Real-Time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras," *IEEE Transactions on Multimedia*, vol. 15, pp. 339-358, 2013.

[18] M. Gailly J.L. Adler. (2013, Oct.) zlib, A Massively Spiffy Yet Delicately Unobtrusive Compression Library. [Online]. "http://www.zlib.net/"

[19] N. Aspert, D. Santa-Cruz, T. Ebrahimi, , "MESH, Measuring Error between Surfaces using hte Haussdorff distance," in *IEEE International conference on media and expo*, 2002.

[20] Institut Telecom. (2013) mymultimediaworld.com. [Online]. "mymultimediaworld.com"

[21] B. Jovanova, M. Preda, F.Preteux, , "MPEG-4 Part 25: A graphics compression framework for XML-based scene graph formats," *Signal Processing: Image Communication*, vol. 24, no. 1/2, p. 101/114, January 2009.