Proceedings Paper:

Jansen van Rensburg, Bianca, Bors, Adrian Gheorghe orcid.org/0000-0001-7838-0021, Puech, William et al. (1 more author) (2023) Simultaneous Watermarking and Draco 3D Object Compression Method. In: IEEE International Conference on Image Processing (ICIP). IEEE , Kuala Lumpur, Malayesia , pp. 725-729.

https://doi.org/10.1109/ICIP49359.2023.10222430

SIMULTANEOUS WATERMARKING AND DRACO 3D OBJECT COMPRESSION METHOD

Bianca Jansen van Rensburg[⋆†] *Adrian G. Bors*[‡] *William Puech*[⋆] *Jean-Pierre Pedeboy*[†]

[⋆] LIRMM, Univ. Montpellier, CNRS, Montpellier, France
[†]STRATEGIES, Rungis, France
[‡]University of York, York, United Kingdom

ABSTRACT

In our present society, 3D objects play an increasingly important role in many different domains, especially with the development of meta environments. Many applications demand large 3D objects, which makes their compression a requirement. Meanwhile, 3D objects are also exposed to various security problems during online storage and transmission. Consequently, security aspects such as copyright and authentication are essential for 3D objects. In this paper, we propose a simultaneous 3D object watermarking and compression method based on Draco. A watermarking step is integrated within Draco, Google's 3D compression method, which is rapidly being universally adopted as standard. The proposed method enables a large bit embedding rate, which can be used for copyright information. The proposed method is also, to the best of our knowledge, the first watermarking method for 3D objects compressed with Draco.

Index Terms— 3D object watermarking, Draco, 3D object compression, 3D object security, multimedia security.

1. INTRODUCTION

Recently, 3D objects are playing an essential role in everyday life, in both private and professional contexts. Throughout their existence, they are regularly transferred over networks and stored on clouds. In many applications 3D objects often consist of millions of vertices, and so storing and sharing them online is expensive, time consuming and not environmentally friendly and, at the same time, they become susceptible to theft. In industry, these 3D objects are often considered as important assets, and their theft can result in a great financial loss, or a leak of trade secrets. In some cases, these 3D objects represent personal information, such as private medical information. For these reasons, it is essential that 3D objects are both compressed and secured.

There are two main categories of methods for securing 3D objects. The first one is 3D object encryption, which consists of protecting the visual confidentiality of 3D objects by rendering them illegible with the use of a secret key. The second one is 3D object watermarking, which aims to embed a secret message in the 3D object in an invisible way. Encryption is generally used when the content of the 3D object needs to be confidential, whereas watermarking is generally used to ensure the 3D object's integrity. Watermarking can be used to embed copyright information, for high capacity data hiding, or to detect whether the 3D object has been subjected to unauthorised alterations or sharing.

Over the years, different solutions for 3D compression have been proposed. In 1999, Rossignac proposed a 3D compression method named Edgebreaker, which compresses the connectivity of 3D objects by means of a depth-first spiralling triangle spanning tree [1]. Notably, in 2014, Google released their 3D object compression method Draco, which is becoming the new industry standard [2]. In 2019, Cao *et al.* performed a survey on 3D point cloud compression [3]. Meanwhile, over the last two decades, several 3D object watermarking methods have been proposed. In 2007, Cho *et al.* proposed a statistical 3D object watermarking method, where the data is embedded by using the distribution of vertex norms [4]. In 2010, Wang *et al.* established a benchmark for 3D object watermarking [5]. Then, in 2013, Bors and Luo proposed a 3D object watermarking method which aims to minimize surface distortion [6]. Later, in 2018, Zhang *et al.* proposed a reversible data hiding method for 3D objects based on prediction-error expansion and sorting [7].

Simultaneous compression and watermarking are very challenging to achieve because they both produce changes in the same domains of the 3D object representation that interfere with each other. In 2009, Abdallah *et al.* proposed a simultaneous compression and watermarking method for 3D objects where data is embedded in the spectral coefficients of a Laplacian sub mesh [8]. Then, Lee *et al.* described a simultaneous reversible watermarking method for progressive 3D object compression, where a watermark is embedded in each level of detail [9]. Although Jansen van Rensburg *et al.* recently proposed the first crypto-compression method for 3D objects [10] which is based on Draco compression, to the best of our knowledge, there are no watermarking methods for Draco compressed 3D objects.

In this paper, we propose a simultaneous watermarking and compression method for 3D objects based on Draco, the industry standard for 3D object compression. We propose in-

---

Thanks to the GDR ISIS, CNRS, for funding this collaboration.

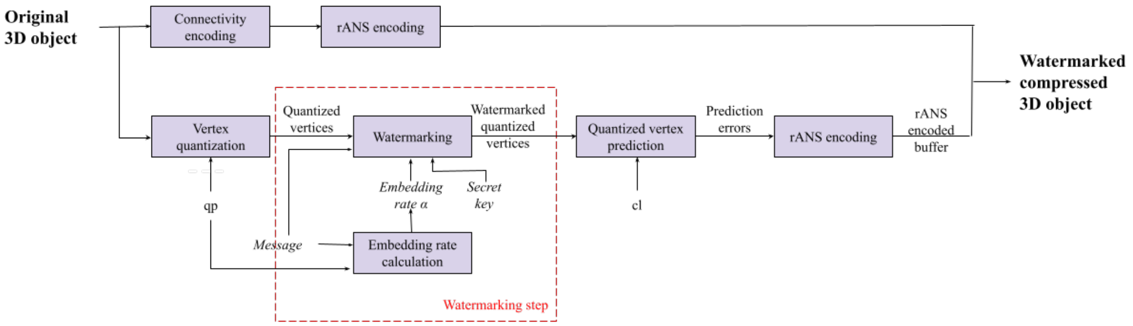


Fig. 1: Overview of the proposed simultaneous Draco 3D object compression and watermarking method.

tegrating the watermarking step between the vertex quantization step and the vertex prediction step of the encoding phase of the Draco 3D object compression method. The watermark is then extracted during Draco's decoding phase, before the vertices are reconstructed.

The rest of this paper is organised as follows. First, in Section 2, we describe the proposed simultaneous watermarking and Draco 3D object compression method. Experimental results are provided in Section 3. Then, in Section 4, we conclude our work.

2. THE PROPOSED SIMULTANEOUS WATERMARKING AND COMPRESSION METHOD

In this section, we outline the proposed simultaneous watermarking and Draco 3D object compression method. Fig. 1 illustrates an overview of the encoding phase of the proposed method. The watermarking step is integrated during the Draco geometry encoding phase after vertices are quantized. After embedding the watermark, the quantized vertices undergo vertex prediction, thus preserving the information embedded in the 3D object. Then, the watermarked compressed 3D object is produced after the entropy encoding. In Section 2.1, we describe the Draco 3D object compression method while in Section 2.2, we detail the watermarking embedding performed during Draco's encoding phase. Then, in Section 2.3 we describe the watermark extraction step, which takes place during Draco's decoding phase.

2.1. 3D Draco compression

As illustrated in Fig. 1, the Draco 3D object compression encoding has two main phases: the connectivity encoding phase which is based on the 3D object compression method Edgebreaker [1], and the geometry encoding phase, which are performed separately. In this paper, we propose integrating the watermark embedding in the geometry encoding phase and we do not interfere with the connectivity encoding phase.

During the geometry encoding phase, the coordinates x, y and z of each vertex are first quantized according to the Draco quantization parameter $qp \in [0, 30]$. Except when $qp = 0$, which signifies that there is no quantization, each coordinate c of each vertex is transformed from a 32 bit floating point to an unsigned integer c' of qp bits. Consequently, the quantized coordinates, $c' \in [0, 2^{qp}]$, are then subjected to a vertex prediction step according to the compression level parameter $cl \in [0, 10]$. We note that the value of qp is a trade-off between the compression rate and the reconstructed 3D object's quality. A range Asymmetric Numeral System (rANS) encoding, based on the entropy encoding scheme Asymmetric Numeral System [11], is performed after the connectivity step and the geometry encoding step, resulting in a compressed 3D object.

2.2. Watermarking and encoding phase

During the geometry encoding phase of the Draco 3D object compression method, we propose integrating a watermarking step, which has no effect on the connectivity encoding phase that is performed separately. More precisely, the watermarking is performed between the vertex quantization step and the vertex prediction step. First, as illustrated in Fig. 1, the length of the message to embed is calculated in order to determine the embedding rate per coordinate, which is limited by the total number of bits per quantized coordinate qp.

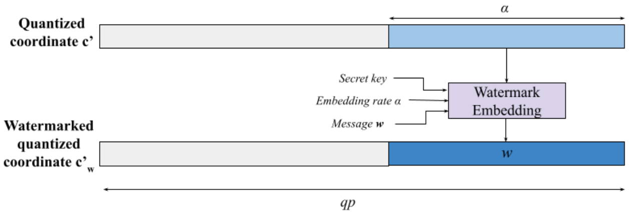


Fig. 2: The watermark embedding step for a single quantized coordinate c'.

Fig. 2 presents the watermark embedding step for a single quantized coordinate c', which has a size of qp bits. Each quantized coordinate c' is watermarked with α bits according to a secret key for message permutation, by means of an LSB substitution:

$$c'_w = \left\lfloor \frac{c'}{2^\alpha} \right\rfloor \times 2^\alpha + w, \qquad (1)$$

where c'_w is the watermarked coordinate, and w the message to be embedded in c'_w, composed of α bits.

After watermarking the quantized coordinates, they undergo the vertex prediction step. The prediction errors of the watermarked quantized vertices are then encoded with the rANS encoding method and a watermarked compressed 3D object in the Draco file format (.drc) is obtained.

2.3. Watermark extraction and decoding

Fig. 3 presents an overview of the simultaneous watermarking and Draco 3D object compression decoding phase. The watermark can be extracted in two places, either after the prediction and vertex reconstruction steps during the decoding phase, or from the reconstructed 3D object in the spatial domain after the entire Draco decoding.

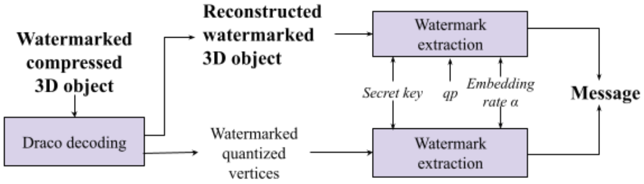


Fig. 3: An overview of the simultaneous Draco 3D object compression and watermarking decoding phase.

First, the watermarked compressed 3D object undergoes a rANS decoding process in order to retrieve the vertex prediction errors. The watermarked quantized vertices are then reconstructed and the watermark can then be retrieved during the decoding step. The message w is extracted by reading the α's LSB of each watermarked reconstructed quantized coordinate, where α corresponds to the embedding rate:

$$w = c'_w \bmod 2^{\alpha}, \tag{2}$$

where c'_w is a watermarked quantized coordinate which has a size of qp bits.

After the watermark is retrieved, the floating point vertices are then reconstructed in order to finalize the Draco decoding process. A watermarked 3D object is then reconstructed, from which the watermark can also be retrieved. In this case the quantization parameter qp is necessary.

3. EXPERIMENTAL RESULTS

In this section, we present the experimental results of our simultaneous watermarking and Draco 3D object compression method. To the best of our knowledge, this is the first paper to propose a joint watermarking and 3D Draco compression method. Therefore, no comparable state-of-the-art method exists. We first apply our proposed method to the 3D object *Bunny*, from the Stanford database [12], as illustrated in Fig. 4. We note that α represents the embedding rate in bits per quantized coordinate. Consequently, with $\alpha = 0$, there is no watermark embedding (Eq. 2).

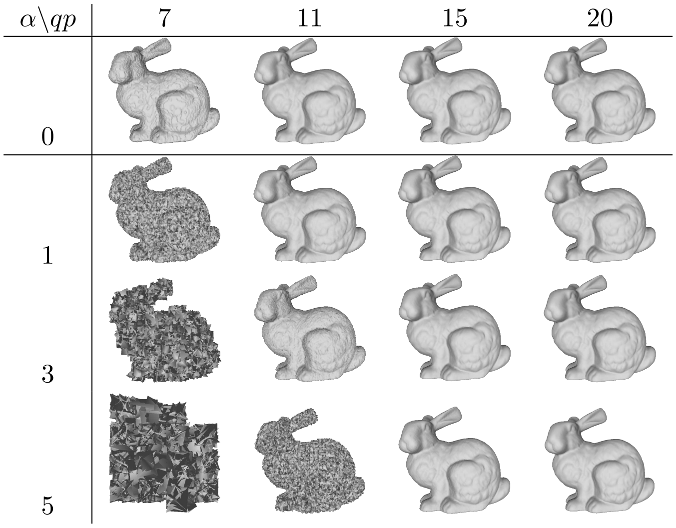
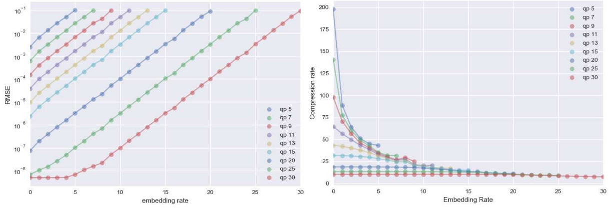


Fig. 4: Visual results of the proposed joint watermarking/3D Draco compression method when applied on *Bunny*, for various payloads α, in bits per quantized coordinate and values of the Draco parameter qp.

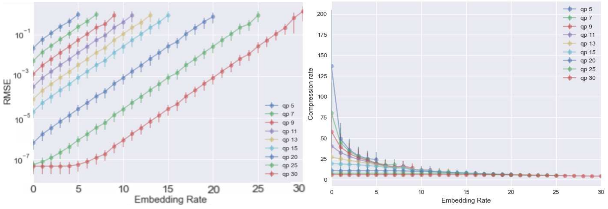
Fig. 4, shows the results of the joint watermarking/3D Draco compressed Bunny object for various combinations of embedding rates α and quantization parameters qp. The embedding rate α must always be lower than qp, since qp represents the number of bits of a quantized coordinate. In other words, when qp increases, α can increase too, as more LSB bits become available for embedding. As the value of α approaches that of qp, the visual degradation increases. We can observe in Fig. 4 that there is no visual degradation when $(qp - \alpha) \geq 7$, which confirms the results from [13] which show that Draco 3D compression does not result in any visual degradation when at least 7 MSBs are unchanged.



(a) RMSE (b) Compression rate

Fig. 5: RMSE and compression rate obtained for the 3D object *Bunny* watermarked and compressed with our proposed method as a function of the embedding rate α for various $qp \in [1, 30]$.

Fig. 5(a) illustrates the RMSE (root mean square error) obtained between the original 3D object *Bunny* and the watermarked and compressed one with our proposed method as a function of the embedding rate α for various $qp \in [1, 30]$. We can observe that the RMSE does not depend only on qp, but also on the relationship between qp and α, *i.e.* the number of MSB left unchanged. For example, when the number of MSB left unchanged is of at least 7, *i.e.* $(qp - \alpha) \geq 7$, the RMSE is negligible, as it always has an order of 10^{-3}, regardless of the value of qp. This explains the visual results from Fig. 4. Fig. 5(b) illustrates the compression rate obtained for the 3D object *Bunny* which has been watermarked and compressed with our proposed method, as a function of the embedding rate α for various $qp \in [1, 30]$. We note that $\alpha \in [0, qp]$, where $\alpha = 0$ corresponds to the original 3D compression method Draco, without watermarking. We observe that when qp increases, the compression rate decreases, this is due to the standard compression of Draco which is illustrated when $\alpha = 0$. This is to be expected, since qp defines the number of bits per coordinate of each vertex after quantization. We also observe that as qp increases, the compression rate reduction becomes less significant as α increases. We can conclude that a trade-off between the quantization parameter qp and the embedding rate α is necessary in order to optimize both the compression rate and the RMSE.



(a) RMSE (b) Compression rate

Fig. 6: Mean RMSE and compression rate obtained for the entire Stanford database [12] watermarked and compressed with our proposed method as a function of the embedding rate α for various $qp \in [1, 30]$.

When applying the proposed method on the entire Stanford database [12], composed of 11 3D objects, the mean RMSE and compression rate as a function of the embedding rate α, for various qp values, is provided in Fig. 6. Fig. 6(a) illustrates the mean RMSE with the standard deviation between the original 3D objects of the Stanford database and their corresponding reconstructed 3D objects which have been simultaneously watermarked and compressed using the proposed method. We note that the standard deviation is negligible, and therefore the RMSE does not greatly vary for different 3D objects. Like for *Bunny*, the RMSE does not depend only on the quantization parameter qp, but on the relationship between qp and α, *i.e.* the number of MSB's left unchanged. In Fig. 6(b), like for *Bunny*, we observe that when qp increases, the compression rate loss when α increases becomes less significant. Therefore, we conclude that the change in the compression rates between different 3D objects is minor, particularly when qp increases.

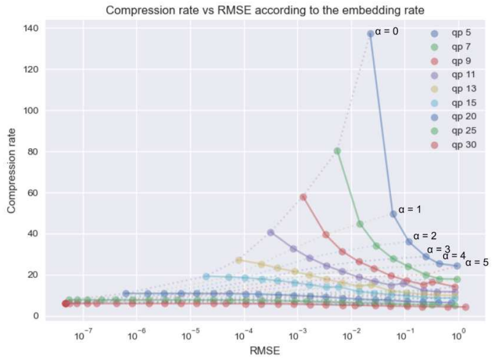


Fig. 7: Mean compression rate as a function of the RMSE for various qp and α, for the entire Stanford database [12].

Like for the standard 3D compression method Draco, there is a trade-off between the compression rate and the distortion. Fig. 7 illustrates the mean compression rate as a function of the mean RMSE for various values of $qp \in [0, 30]$ for the Stanford database. We observe that changing the quantization parameter qp has a great impact on the compression rate and a minor impact on the RMSE. Therefore, the choice of qp should be made according to the desired embedding rate α. For example, if the desired embedding rate is $\alpha = 1$ bit per coordinate (3 bits per vertex), and the user wishes to minimize the distortion, then $qp = 15$ is recommended, as the average RMSE is 5.22×10^{-5}. If, for another example, the desired embedding rate α is 3 bits per coordinate (9 bits per vertex), then a good trade-off between the compression rate and the distortion is $qp = 11$. Indeed, with $qp = 11$, which is the default parameter for Draco given by Google, we achieve an average compression rate of 24.38 and an average RMSE of 3.47×10^{-3}. In conclusion, for these two examples ($qp = 15, \alpha = 1$ and $qp = 11, \alpha = 3$), we observe in Fig. 4 that the reconstructed 3D objects have a high quality without visual degradation.

4. CONCLUSION

In this paper, we proposed the first simultaneous watermarking and Draco 3D object compression method. The watermarking is integrated during the geometry encoding process of the Draco 3D object compression method. The watermark can then be extracted either from the Draco format file (.drc) during the Draco decoding process, or after the 3D object is decoded and reconstructed, as the reconstruction still contains the watermark. Experimental results show that we can achieve a large watermark embedding rate, which can be used for embedding a significant amount of information besides that related to the copyright. We have proposed an optimal value for the Draco and watermarking parameters in different scenarios. We have also tested the proposed method on a large database. In future work, we want to integrate the watermark, while trying to reduce the distortion as much as possible and keeping the original compression rate.

5. REFERENCES

[1] J. Rossignac, “3D compression made simple: Edgebreaker with ZipandWrap on a corner-table,” in *Proceedings International Conference on Shape Modeling and Applications*, 2001, pp. 278–283.

[2] Google, “Draco 3D graphics compression,” 2014.

[3] C. Cao, M. Preda, and T. Zaharia, “3D point cloud compression: A survey,” in *The 24th International Conference on 3D Web Technology*, 2019, pp. 1–9.

[4] J.-W. Cho, R. Prost, and H.-Y.Jung, “An Oblivious Watermarking for 3-D Polygonal Meshes Using Distribution of Vertex Norms,” *IEEE Transactions on Signal Processing*, vol. 55, pp. 142 – 155, 02 2007.

[5] K. Wang, G. Lavoué, F. Denis, A. Baskurt, and X. He, “A Benchmark for 3D Mesh Watermarking,” in *2010 Shape Modeling International Conference*, 2010, pp. 231–235.

[6] A. G. Bors and M. Luo, “Optimized 3D Watermarking for Minimal Surface Distortion,” *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1822–1835, 2013.

[7] Q. Zhang, X. Song, T. Wen, and C. Fu, “Reversibility improved data hiding in 3D mesh models using prediction-error expansion and sorting,” *Measurement*, vol. 135, pp. 738–746, 2019.

[8] E. Abdallah, A. Hamza, and P. Bhattacharya, “Watermarking 3D models using spectral mesh compression,” *Signal, Image and Video Processing*, vol. 3, pp. 375–389, 10 2009.

[9] H. Lee, Ç. Dikici, G. Lavoué, and F. Dupont, “Joint reversible watermarking and progressive compression of 3D meshes,” *The Visual Computer*, vol. 27, pp. 781–792, 2011.

[10] B. Jansen van Rensburg, W. Puech, and J.-P. Pedeboy, “The First Draco 3D Object Crypto-Compression Scheme,” *IEEE Access*, vol. 10, pp. 10566–10574, 2022.

[11] J. Duda, K. Tahboub, N. J. Gadgil, and E. J. Delp, “The use of asymmetric numeral systems as an accurate replacement for Huffman coding,” in *Picture Coding Symposium (PCS)*, 2015, pp. 65–69.

[12] M. Levoy, J. Gerth, B. Curless, and K. Pull, “The Stanford 3D scanning repository,” *URL http://graphics.stanford.edu/data/3Dscanrep/*, vol. 5, no. 10, 2005.

[13] B. Jansen van Rensburg, W. Puech, and J.-P. Pedeboy, “Draco-Based Selective Crypto-Compression Method of 3D objects,” in *IEEE IPTA 2022*, 2022, pp. 1–6.