Optimized Dynamic Point Cloud Compression OPT-PCC

Grant Agreement ID: 836192

Deliverable D3

Report on the bit allocation solution

Authors:	Hui Yuan, Raouf Hamzaoui, Ferrante Neri, Shengxiang Yang
Version:	V2.0
Date:	25/08/2021
Classification:	Public
Contract Start Date:	23/11/2020
Coordinator:	De Montfort University
File Name:	OPT-PCC D3.docx

MSCA-IF-2018 - Individual Fellowships Project funded by the European Commission under:

H2020-EU.1.3.2. - Nurturing excellence by means of cross-border and cross-sector mobility

Executive Summary

Point clouds are representations of three-dimensional (3D) objects in the form of a sample of points on their surface. Point clouds are receiving increased attention from academia and industry due to their potential for many important applications, such as real-time 3D immersive telepresence, automotive and robotic navigation, as well as medical imaging. Compared to traditional video technology, point cloud systems allow free viewpoint rendering, as well as mixing of natural and synthetic objects. However, this improved user experience comes at the cost of increased storage and bandwidth requirements as point clouds are typically represented by the geometry and colour (texture) of millions up to billions of 3D points. For this reason, major efforts are being made to develop efficient point cloud compression schemes. However, the task is very challenging, especially for dynamic point clouds (sequences of point clouds), due to the irregular structure of point clouds (the number of 3D points may change from frame to frame, and the points within each frame are not uniformly distributed in 3D space). To standardize point cloud compression (PCC) technologies, the Moving Picture Experts Group (MPEG) launched a call for proposals in 2017. As a result, three point cloud compression technologies were developed: surface point cloud compression (S-PCC) for static point cloud data, video-based point cloud compression (V-PCC) for dynamic content, and LIDAR point cloud compression (L-PCC) for dynamically acquired point clouds. Later, L-PCC and S-PCC were merged under the name geometry-based point cloud compression (G-PCC). The aim of the OPT-PCC project is to develop algorithms that optimise the rate-distortion performance [i.e., minimize the reconstruction error (distortion) for a given bit budget] of V-PCC. The objectives of the project are to:

1. O1: build analytical models that accurately describe the effect of the geometry and colour quantization of a point cloud on the bit rate and distortion;
2. O2: use O1 to develop fast search algorithms that optimise the allocation of the available bit budget between the geometry information and colour information;
3. O3: implement a compression scheme for dynamic point clouds that exploits O2 to outperform the state-of-the-art in terms of rate-distortion performance. The target is to reduce the bit rate by at least 20% for the same reconstruction quality;
4. O4: provide multi-disciplinary training to the researcher in algorithm design, metaheuristic optimisation, computer graphics, media production, and leadership and management skills.

This deliverable reports on the work undertaken in this project to achieve objective O2. Section 1 introduces the rate-distortion optimization problem for V-PCC. Section 2 reviews previous work. Section 3 presents our fast search algorithms. Section 4 gives experimental results. Section 5 gives our conclusions.

Table of Contents

1 Introduction

A static point cloud is a representation of a three-dimensional (3D) object, where in addition to the spatial coordinates of a sample of points on the surface of the object, attributes such as color, reflectance, transparency, and normal direction may be used (Fig. 1). A dynamic point cloud consists of several successive static point clouds. Each point cloud in the sequence is called a frame. Point clouds are receiving increased attention due to their potential for immersive video experience applications such as virtual reality, augmented reality, and immersive telepresence.

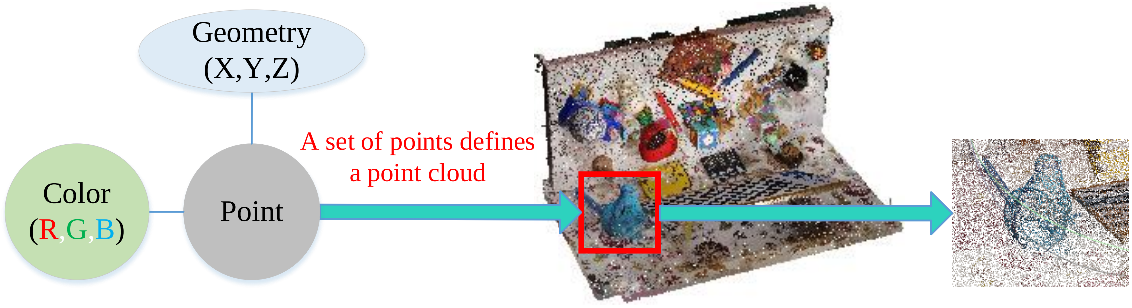

Fig. 1. Point cloud representation with color used as an attribute.

To get a high-quality representation of a 3D object as a point cloud, a huge amount of data is required. For example, in a point cloud consisting of 1 million points, each dimension of the geometry (resp. color) information is usually represented by 12 bits (resp. 8 bits), resulting in a total volume of 60 Mbits. For a dynamic point cloud with a frame rate of 25 frames per second (fps), the required bit rate is therefore 1500 Mbps, which is beyond the bandwidth capacity of current networks.

To compress point clouds efficiently, the Moving Picture Experts Group (MPEG) launched in January 2017 a call for proposals for point cloud compression technology. As a result, two point cloud compression standards were developed: video-based point cloud compression (V-PCC) [1] for point sets with a relatively uniform distribution of points and geometry-based point cloud compression (G-PCC) [2] for more sparse distributions.

In this project, we focus on V-PCC for dynamic point clouds. In V-PCC, the input point cloud is first decomposed into a set of patches, which are independently mapped to a two-dimensional grid of uniform blocks. This mapping is then used to store the geometry and color information as one geometry video and one color video. Next, the generated geometry video and color video are compressed separately with a video coder, e.g., H.265/HEVC [3]. Finally, the geometry and color videos, together with metadata (occupancy map for the two-dimensional grid, auxiliary patch, and block information) are multiplexed to generate the bit stream (Fig. 2). In the video coding step, compression is achieved with quantization, which is determined by a quantization step or, equivalently, a quantization parameter (QP), see Appendix A.

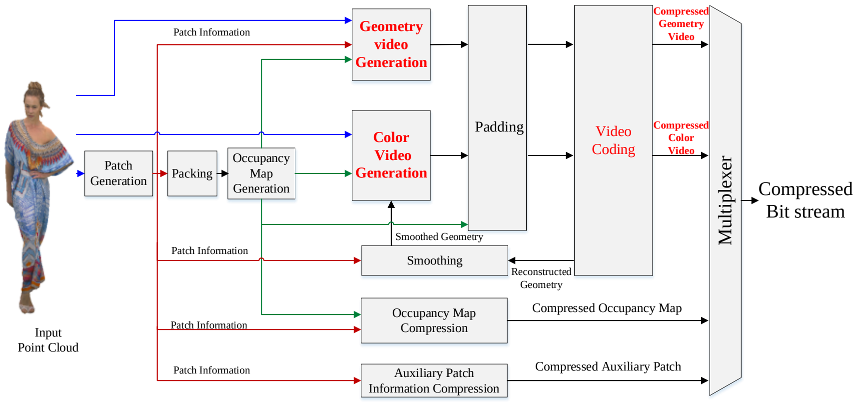

Fig. 2. V-PCC encoder test model [1].

Given a dynamic point cloud consisting of N frames, an optimal encoding can be obtained by determining for each frame i ($i = 1, \dots, N$) the geometry quantization step $Q_{g,i} \in \{q_0, \dots, q_{M-1}\}$ and colour quantization step $Q_{c,i} \in \{q_0, \dots, q_{M-1}\}$ that minimize the distortion subject to a constraint R_T on the bit budget. This can be formulated as the multi-objective optimization problem

$$\min_{\boldsymbol{Q}_g,\boldsymbol{Q}_c} \left[D_g(\boldsymbol{Q}_g, \boldsymbol{Q}_c), D_c(\boldsymbol{Q}_g, \boldsymbol{Q}_c)\right] \qquad (1)$$
$$s.t. \quad R(\boldsymbol{Q}_g, \boldsymbol{Q}_c) \le R_T,$$

where $\boldsymbol{Q}_g = (Q_{g,1}, Q_{g,2,\dots,}Q_{g,N})$, $\boldsymbol{Q}_c = (Q_{c,1}, Q_{c,2,\dots,}Q_{c,N})$, $D_g(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ is the geometry distortion, $D_c(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ is the color distortion, and $R(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ is the total number of bits. Here $D_g(\boldsymbol{Q}_g, \boldsymbol{Q}_c) = \frac{1}{N}\sum_{i=1}^{N} D_{g,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ and $D_c(\boldsymbol{Q}_g, \boldsymbol{Q}_c) = \frac{1}{N}\sum_{i=1}^{N} D_{c,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$,where $D_{g,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ and $D_{c,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ are the geometry and color distortions of the i th frame, respectively. Similarly, $R(\boldsymbol{Q}_g, \boldsymbol{Q}_c) = R_g(\boldsymbol{Q}_g, \boldsymbol{Q}_c) + R_c(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$, where $R_g(\boldsymbol{Q}_g, \boldsymbol{Q}_c) = \sum_{i=1}^{N} R_{g,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ is the number of bits for the geometry information, $R_{g,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ is the number of bits for the geometry information in the ith frame, $R_c(\boldsymbol{Q}_g, \boldsymbol{Q}_c) = \sum_{i=1}^{N} R_{c,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ is the number of bits for the color information, and $R_{c,i}(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ is the number of bits for the color information in the ith frame. In practice, problem (1) is scalarized as

$$\min_{\boldsymbol{Q}_g,\boldsymbol{Q}_c} \left[D(\boldsymbol{Q}_g, \boldsymbol{Q}_c) = \omega D_c(\boldsymbol{Q}_g, \boldsymbol{Q}_c) + (1-\omega) D_g(\boldsymbol{Q}_g, \boldsymbol{Q}_c)\right] \qquad (2)$$
$$s.t. \quad R(\boldsymbol{Q}_g, \boldsymbol{Q}_c) \le R_T,$$

where $\omega \in [0,1]$ is a weighting factor that sets the relative importance of the geometry and color distortions. As the number of possible solutions is M^{2N}, solving the problem with exhaustive search is not feasible when M or N is large as the computation of the distortion and the number of bits requires encoding and decoding the point cloud, which is very time consuming.

In the latest MPEG V-PCC test model [4], the QPs for geometry and color are selected manually: one chooses the QPs of the first frame, and the QPs of the following frames in the same group of frames are set according to a fixed offset. In this deliverable, we propose two methods to solve problem (2). The first one uses the rate and distortion models proposed in Deliverable D2 [5]. The second one uses the actual rate and distortion. The two methods apply a variant of differential evolution (DE) [6] to solve the problem.

2 Related Work

Only a small number of works [7,8,9] have proposed rate and distortion models for point cloud compression. In [7], the focus is on the point cloud library (PCL) platform [10] for the compression of static point clouds. This platform uses an octree decomposition for geometry compression and JPEG for color compression. Analytical models that describe the relationship between the encoding parameters (the maximum octree level and the JPEG quality factor) and the color distortion D_c and bitrate R are derived with statistical analysis. Let L be the maximum octree level and let J be the JPEG quality factor. The color distortion is modeled as $D_c = sJ^pL^q$, where s, p, q are model parameters. On the other hand, the bitrate is modeled as $\ln R = aLJ + bL + c$, where a, b, c are model parameters. Then, the models are used to formulate the rate-distortion optimization problem as a constrained convex optimization problem, and an interior point method is applied to solve it. In [8], a similar approach is applied to V-PCC for dynamic point clouds. First, distortion and rate models for the geometry information and color information are derived as follows: $D_g = \alpha_g Q_{g,1} + \delta_g$, $D_c = \alpha_c Q_{c,1} + \beta_c Q_{g,1} + \delta_c$, $R_g = \gamma_g Q_{g,1}^{\theta_g}$, $R_c = \gamma_c Q_{c,1}^{\theta_c}$, where α_g , $\delta_g, \alpha_c, \beta_c, \delta_c, \gamma_g, \theta_g, \gamma_c, \theta_c$ are model parameters. Then, an interior point method is used to minimize the weighted sum of the distortions subject to a constraint on the bitrate. One limitation of this work is that the distortion and rate models are functions of the quantization steps of the geometry and color information of the first frame only. Thus, the models are only suitable when the quantization steps of the following frames are set according to the default settings of the V-PCC test model and are not appropriate for the general rate-distortion optimization problem (2). In [9], a static point cloud is partitioned into seven regions such that the first six regions correspond to the six patches with the largest area in the six projection planes, and the seventh region consists of all other patches. Then, the geometry and color quantization steps of the video sequences corresponding to each region are optimized separately using the analytical models in [8] for the distortion and bitrate. Here too, only the quantization steps of the first frame are considered.

3 Bit Allocation Solution

3.1 Distortion and Rate models

According to deliverable D2 [5], the distortion model of frame i can be written as

$$\begin{cases} D_{g,i} = \alpha_{g,i} Q_{g,i} + \delta_{g,i} \\ D_{c,i} = \alpha_{c,i} Q_{c,i} + \beta_{c,i} Q_{g,i} + \delta_{c,i}, \end{cases} \tag{3}$$

where $\alpha_{g,i}$, $\delta_{g,i}$, $\alpha_{c,i}$, $\beta_{c,i}$, and $\delta_{c,i}$ are model parameters. The overall distortion of a GOP is then modeled as

$$D = \frac{1}{4}\left(\sum_{i=1}^{4} \omega D_{g,i} + (1-\omega) D_{c,i}\right) \tag{4}$$

The rate models of the first frame can be modeled as

$$\begin{cases} R_{g,1} = \gamma_{g,1} Q_{g,1}^{\theta_{g,1}} \\ R_{c,1} = \gamma_{c,1} Q_{c,1}^{\theta_{c,1}} \end{cases} \tag{5}$$

where $\gamma_{g,1}$, $\gamma_{c,1}$, $\theta_{g,1}$, and $\theta_{c,1}$ are model parameters. By taking the inter-frame dependency into account, the rate model of the second, third, and the fourth frames can be written as

$$\begin{cases} R_{g,2} = (\varphi_{g,(1,2)} \cdot Q_{g,1} + 1)\gamma_{g,2} Q_{g,2}^{\theta_{g,2}} \\ R_{c,2} = (\varphi_{c,(1,2)} \cdot Q_{c,1} + 1)\gamma_{c,2} Q_{c,2}^{\theta_{c,2}} \end{cases} \tag{6}$$

$$\begin{cases} R_{g,3} = \prod_{i=1}^{2} (\varphi_{g,(i,i+1)} \cdot Q_{g,i} + 1)\, \gamma_{g,3} Q_{g,3}^{\theta_{g,3}} \\ R_{c,3} = \prod_{i=1}^{2} (\varphi_{c,(i,i+1)} \cdot Q_{c,i} + 1)\, \gamma_{c,3} Q_{c,3}^{\theta_{c,3}} \end{cases} \tag{7}$$

$$\begin{cases} R_{g,4} = \prod_{i=1}^{3} (\varphi_{g,(i,i+1)} \cdot Q_{g,i} + 1)\, \gamma_{g,4} Q_{g,4}^{\theta_{g,4}} \\ R_{c,4} = \prod_{i=1}^{3} (\varphi_{c,(i,i+1)} \cdot Q_{c,i} + 1)\, \gamma_{c,4} Q_{c,4}^{\theta_{c,4}} \end{cases} \tag{8}$$

where $\varphi_{g,(1,2)}$ and $\varphi_{c,(1,2)}$ are the impact factors of the first frame on the second frame, $\varphi_{g,(i,i+1)}$ and $\varphi_{c,(i,i+1)}$ $(i = 2,3)$ are the impact factors of the i-th frame on the $(i+1)$-th one, $\gamma_{g,2}$, $\gamma_{c,2}$, $\theta_{g,2}$, $\theta_{c,2}$, $\gamma_{g,3}$, $\gamma_{c,3}$, $\theta_{g,3}$, $\theta_{c,3}$, $\gamma_{g,4}$, $\gamma_{c,4}$, $\theta_{g,4}$, and $\theta_{c,4}$ are model parameters. Finally, we use (5), (6), (7) and (8) to build the rate model as $R=\sum_{i=1}^{4} R_{g,i} + R_{c,i}$.

3.2 Model Parameters

To determine the parameters of the distortion models, we first encode the point cloud for three different sets of quantization steps $(\boldsymbol{Q}_g, \boldsymbol{Q}_c)$ and compute the corresponding actual distortions and number of bits. Next, we solve the resulting system of equations to find $\alpha_{g,i}$, $\delta_{g,i}$, $\alpha_{c,i}$, $\beta_{c,i}$, $\delta_{c,i}$ $(i = 1, \ldots, 4)$. To determine the parameters of the rate models, we encode the point cloud for eight more sets of quantization steps and use linear regression in (5), (6), (7), and (8) to estimate the parameters $\gamma_{g,i}, \theta_{g,i}, \gamma_{c,i}, \theta_{c,i}$ $(i = 1, \ldots, 4)$. Finally, the impact factors $\varphi_{g,(1,2)}$, $\varphi_{g,(2,3)}$, $\varphi_{g,(3,4)}$, $\varphi_{c,(1,2)}$ $\varphi_{c,(2,3)}$, and $\varphi_{c,(3,4)}$, are empirically set to

$$\begin{cases} \varphi_{g,(1,2)} = \varphi_{c,(1,2)} = 0.004 \\ \varphi_{g,(2,3)} = \varphi_{c,(2,3)} = 0.0015 \\ \varphi_{g,(3,4)} = \varphi_{c,(3,4)} = 0.0010. \end{cases} \tag{9}$$

The QP settings to determine the model parameters are shown in Table 1.

Table 1. QP settings for pre-coding.

Model parameters	$QP_{g,1}$	$QP_{g,2}$	$QP_{g,3}$	$QP_{g,4}$	$QP_{c,1}$	$QP_{c,2}$	$QP_{c,3}$	$QP_{c,4}$
$\alpha_{g,1}, \delta_{g,1}; \alpha_{g,2}, \delta_{g,2}; \alpha_{g,3}, \delta_{g,3}; \alpha_{g,4}, \delta_{g,4}; \alpha_{c,1}, \beta_{c,1}, \delta_{c,1}; \alpha_{c,2}, \beta_{c,2}, \delta_{c,2}; \alpha_{c,3}, \beta_{c,3}, \delta_{c,3}; \alpha_{c,4}, \beta_{c,4}, \delta_{c,4}$	30	30	30	30	40	40	40	40
	36	36	36	36	30	30	30	30
	38	38	38	38	28	28	28	28
$\gamma_{g,1}, \theta_{g,1}; \gamma_{c,1}, \theta_{c,1}; \gamma_{g,2}, \theta_{g,2}; \gamma_{c,2}, \theta_{c,2}; \gamma_{g,3}, \theta_{g,3}; \gamma_{c,3}, \theta_{c,3}; \gamma_{g,4}, \theta_{g,4}; \gamma_{c,4}, \theta_{c,4}$	30	30	30	30	40	40	40	40
	36	36	36	36	30	30	30	30
	38	38	38	38	28	28	28	28
	17	25	33	41	17	25	33	41
	33	25	33	41	33	25	33	41
	17	41	33	41	17	41	33	41
	17	25	49	41	17	25	49	41
	19	24	29	34	19	24	29	34
	34	24	40	37	34	24	40	37
	27	41	37	45	27	41	37	45
	27	17	37	45	27	17	37	45

3.3 Solution

Model-based DE solution

To solve the rate-distortion optimization problem (2), we apply a DE variant to the analytical models presented in Section 3.1. Starting from a population of randomly selected solutions, DE generates for each solution an offspring by perturbing another solution from the population with a scaled difference of two randomly selected solutions from the population. If the offspring is a better solution than the parent, the parent is replaced by the offspring. This procedure is repeated for a given number of iterations. One of the advantages of DE is that it has only three control parameters: the population size NP, a scaling factor μ that scales the difference of the two randomly selected solutions, and a crossover rate CR that controls the number of parents that may be replaced.

The details of the implemented algorithm are as follows. A candidate solution (agent) for problem (2) is denoted by $\boldsymbol{x} = (\boldsymbol{Q}_g, \boldsymbol{Q}_c) = (x_1, x_2, \dots, x_{2N})$.

- Choose a population size NP, an interval I for the scaling factor, and a number of iterations n.
- Build a population of NP agents $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(NP)}$ such that each component ${x_i}^{(j)}, i = 1, \dots, 2N; j = 1, \dots, NP,$ is randomly chosen in the set of quantization steps $\{q_0, \dots, q_{M-1}\}$ and $R(\boldsymbol{x}^{(j)}) \leq R_T$ for $j = 1, \dots, NP$.
- FOR $k = 1$ to n
  - If $k < \frac{2}{3}n$, set the crossover rate to $CR = 0.9$; otherwise, set $CR = 0.1$;
    - FOR $j = 1$ to NP

      Step 1: Select randomly from the population three different agents $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ that are also different from $\boldsymbol{x}^{(j)}$

      Step 2: Select randomly an index r such that $1 \leq r \leq 2N$

      Step 3: Compute a candidate new agent $\boldsymbol{y}^{(j)}$ as follows:
      - For each $i \in \{1, \dots, 2N\}$, choose a random number r_i according to a uniform distribution in (0,1). Choose a scaling factor w randomly in I.
      - If $r_i \leq CR$ or $i = r$, then set ${y_i}^{(j)} = a_i + w \times (b_i - c_i)$; otherwise, set ${y_i}^{(j)} = {x_i}^{(j)}$

- If ${y_i}^{(j)} < q_0$, set ${y_i}^{(j)} = q_0$. If ${y_i}^{(j)} > q_{M-1}$, set ${y_i}^{(j)} = q_{M-1}$.

Step 4: If $D\left(\boldsymbol{y}^{(j)}\right) < D\left(\boldsymbol{x}^{(j)}\right)$ and $R(\boldsymbol{y}^{(j)}) \le R_T$, note j.

END FOR

FOR $j = 1$ to NP, replace $\boldsymbol{x}^{(j)}$ by $\boldsymbol{y}^{(j)}$ if j was noted in Step 4.

END FOR

END FOR

- Select the agent from the population that gives the lowest distortion D and round the components of this agent to the nearest values in the set $\{q_0, \dots, q_{M-1}\}$.

Our implementation differs from standard DE in three ways. First, we decrease the crossover rate CR at runtime to increase the exploitation pressure. As shown in [11], DE frameworks are prone to stagnate. That is, the population of the algorithm is diverse and yet searches in the decision space without succeeding at generating a solution outperforming the best individual of the population. This stagnation can be mitigated by exploiting the search directions available in the DE population [12]. A reduction in the crossover rate CR makes the offspring similar to the generating parent and thus exploits the available genotypes. Second, in accordance with [13], we select the scaling factor μ randomly to retain population diversity as the search progresses. Our experiments show that this randomization is beneficial.

Another way of solving problem (2) is to use conventional non-evolutionary constrained nonlinear optimization algorithms. However, when the problem is not convex, such algorithms are only guaranteed to find local minima and are very sensitive to the starting point of the algorithm.

Encoding-based DE solution

An alternative to the model-based optimization described in the previous section is to apply our DE variant to the actual rate and distortion functions. This approach is computationally intensive since we must encode the point cloud each time the distortion and rate are evaluated. First, we rewrite problem (2) as

$$\min_{\boldsymbol{x}} \left[D(\boldsymbol{x}) = \omega D_g(\boldsymbol{x}) + (1-\omega)D_c(\boldsymbol{x})\right] \quad (10)$$
$$s.t. \quad R(\boldsymbol{x}) \le R_T,$$

where $\boldsymbol{x} = (x_1, x_2, \dots, x_{2N}) = \left(Q_{g,1}, Q_{g,2}, \dots, Q_{g,N}, Q_{c,1}, Q_{c,2}, \dots, Q_{c,N}\right)$. Let f be the one-to-one correspondence that maps a quantization step $x \in (q_0, \dots, q_{M-1})$ to a QP, i.e., $f(x) \in \{0, \dots, M-1\}$. Each solution $\boldsymbol{x}$ of problem (10) can be mapped in a unique way to a vector $\boldsymbol{X} = F(\boldsymbol{x}) = \left(f(x_1), f(x_2), \dots f(x_{2N})\right)$. The algorithm proceeds as follows.

Parameter setting: Choose a population size NP, an interval I for the scaling factor μ, and a number of iterations n.

Initialization: Build a population of NP vectors $\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(NP)}$ by randomly choosing each component $\mathrm{X}_i^j, i = 1, \dots, 2N; j = 1, \dots, NP$, in the set $\{0,1, \dots M-1\}$.

for $k = 1$ to n **do**

if $k < 2n/3$ **then**

Set the crossover rate $\boldsymbol{CR}$ to 0.9

else

$\boldsymbol{CR}$ = 0.1

end if

for $j = 1$ to NP **do**

Step 1: Select randomly from the population three different vectors $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ that are also different from $\boldsymbol{X}^{(j)}$

Step 2: Select randomly an integer r such that $1 \leq r \leq 2N$.

Step 3: Build a candidate vector $\boldsymbol{Y}^{(j)}$ as follows. For each $i \in \{1, \ldots, 2N\}$, choose a random number r_i according to a uniform distribution in (0.1, 0.9). Choose a scaling factor μ randomly in *I*. If $r_i \leq CR$ or $i = r$, then set $\boldsymbol{Y}_i^{(j)} = round(A_i + \mu(B_i - C_i))$; otherwise, set $\boldsymbol{Y}_i^{(j)} = \boldsymbol{X}_i^{(j)}$**.** Here, round means rounding to the nearest integer in the set $\{0, \ldots, M-1\}$.

Step 4: If $D\left(F^{-1}\left(\boldsymbol{Y}^{(j)}\right)\right) < D\left(F^{-1}\left(\boldsymbol{X}^{(j)}\right)\right)$ and $R\left(F^{-1}\left(\boldsymbol{Y}^{(j)}\right)\right) \leq R_T$, mark j.

end for

for $j = 1$ to NP **do**

replace $\boldsymbol{X}^{(j)}$ by $\boldsymbol{Y}^{(j)}$ if *j* was marked in Step 4.

end for

end for

Output: Select the vector from the population that gives the lowest distortion D.

Note that we use rounding inside the iterations so that the actual distortion and rate can be computed.

4 Experimental Results

In this section, we use our two DE-based optimization algorithms to solve the rate-distortion optimization problem for the encoding of a group of frames consisting of four frames with the low-delay configuration of V-PCC test model V12 [4]. Tables 2 and 3 show the geometry and color QPs of each frame for various point clouds and target bitrates. The bitrates are expressed in kilobits per million points (kbpmp). The tables show that the QPs computed by the proposed algorithms can be significantly different from the ones computed with the method in [8].

Table 2. Comparison between the method in [8] and the model-based DE solution

Point Cloud	Target bitrate (kbpmp)	(QP_g1,QP_g2,QP_g3,QP_g4,QP_c1,QP_c2,QP_c3,QP_c4) [8]	Model-based
soldier	65	(36, 36, 36, 36, 38, 38, 38, 38)	(38, 38, 37, 39, 40, 38, 38, 38)
	125	(30, 30, 30, 30, 34, 34, 34, 34)	(32, 32, 31, 33, 36, 34, 33, 33)
	165	(28, 28, 28, 28, 32, 32, 32, 32)	(30, 30, 29, 30, 33, 32, 31, 31)
	210	(26, 26, 26, 26, 30, 30, 30, 30)	(28, 28, 27, 28, 32, 31, 29, 30)
	265	(24, 24, 24, 24, 28, 28, 28, 28)	(26, 27, 25, 26, 30, 29, 28, 28)
	365	(22, 22, 22, 22, 26, 26, 26, 26)	(24, 24, 22, 23, 28, 27, 26, 26)
queen	65	(30, 30, 30, 30, 40, 40, 40, 40)	(33, 31, 27, 31, 40, 39, 38, 40)
	125	(24, 24, 24, 24, 34, 34, 34, 34)	(27, 26, 20, 25, 35, 34, 34, 35)
	165	(22, 22, 22, 22, 32, 32, 32, 32)	(25, 24, 18, 23, 33, 32, 31, 33)
	210	(22, 22, 22, 22, 30, 30, 30, 30)	(22, 22, 16, 22, 31, 31, 30, 32)
	265	(22, 22, 22, 22, 28, 28, 28, 28)	(21, 20, 13, 20, 30, 29, 28, 30)
	365	(22, 22, 22, 22, 24, 24, 24, 24)	(18, 18, 11, 17, 28, 27, 26, 28)
loot	65	(36, 36, 36, 36, 36, 36, 36, 36)	(37, 37, 38, 38, 37, 36, 35, 35)
	125	(30, 30, 30, 30, 30, 30, 30, 30)	(32, 31, 31, 31, 32, 31, 30, 30)
	165	(26, 26, 26, 26, 28, 28, 28, 28)	(29, 28, 28, 29, 31, 29, 27, 28)
	210	(24, 24, 24, 24, 28, 28, 28, 28)	(27, 26, 26, 27, 29, 28, 26, 27)
	265	(22, 22, 22, 22, 26, 26, 26, 26)	(25, 24, 24, 25, 27, 26, 24, 25)
	365	(22, 22, 22, 22, 22, 22, 22, 22)	(22, 21, 21, 22, 25, 24, 21, 23)
basketballplayer	30	(40, 40, 40, 40, 40, 40, 40, 40)	(38, 41, 43, 43, 40, 40, 41, 41)
	65	(32, 32, 32, 32, 34, 34, 34, 34)	(31, 34, 36, 34, 34, 35, 34, 34)
	125	(26, 26, 26, 26, 28, 28, 28, 28)	(25, 28, 29, 28, 30, 30, 29, 29)
	165	(24, 24, 24, 24, 26, 26, 26, 26)	(23, 27, 27, 26, 28, 28, 27, 27)
	210	(22, 22, 22, 22, 24, 24, 24, 24)	(21, 24, 25, 23, 27, 26, 25, 26)
	265	(22, 22, 22, 22, 22, 22, 22, 22)	(20, 22, 22, 21, 25, 25, 24, 24)
redandblack	90	(40, 40, 40, 40, 40, 40, 40, 40)	(37, 43, 41, 44, 40, 41, 40, 41)
	180	(32, 32, 32, 32, 34, 34, 34, 34)	(31, 36, 33, 36, 34, 34, 33, 34)
	270	(28, 28, 28, 28, 30, 30, 30, 30)	(28, 32, 28, 32, 31, 31, 29, 30)
	360	(26, 26, 26, 26, 28, 28, 28, 28)	(25, 30, 25, 28, 29, 28, 27, 27)
	480	(24, 24, 24, 24, 24, 24, 24, 24)	(23, 26, 22, 25, 26, 26, 24, 25)
	640	(22, 22, 22, 22, 22, 22, 22, 22)	(19, 24, 19, 22, 24, 24, 22, 22)
longdress	180	(30, 30, 30, 30, 38, 38, 38, 38)	(31, 31, 31, 34, 38, 38, 38, 39)
	270	(26, 26, 26, 26, 34, 34, 34, 34)	(28, 27, 28, 30, 35, 35, 35, 36)
	360	(24, 24, 24, 24, 32, 32, 32, 32)	(26, 24, 25, 27, 33, 33, 33, 34)
	480	(22, 22, 22, 22, 30, 30, 30, 30)	(24, 22, 22, 24, 31, 31, 31, 32)
	640	(22, 22, 22, 22, 28, 28, 28, 28)	(21, 20, 20, 23, 29, 29, 29, 30)
	840	(22, 22, 22, 22, 26, 26, 26, 26)	(19, 17, 16, 20, 27, 27, 28, 28)

Table 3. Comparison between the method in [8] and the encoding-based DE solution

Point Cloud	Target bitrate (kbpmp)	(QP_g1,QP_g2,QP_g3,QP_g4,QP_c1,QP_c2,QP_c3,QP_c4) [8]	(QP_g1,QP_g2,QP_g3,QP_g4,QP_c1,QP_c2,QP_c3,QP_c4) Encoding-based
soldier	65	(36, 36, 36, 36, 38, 38, 38, 38)	(34, 42, 40, 44, 35, 43, 42, 44)
	165	(28, 28, 28, 28, 32, 32, 32, 32)	(24, 30, 31, 33, 29, 37, 33, 37)
	265	(24, 24, 24, 24, 28, 28, 28, 28)	(19, 28, 24, 27, 26, 33, 30, 34)
	365	(22, 22, 22, 22, 26, 26, 26, 26)	(18, 22, 20, 21, 25, 30, 27, 30)
queen	65	(30, 30, 30, 30, 40, 40, 40, 40)	(25, 36, 27, 33, 36, 40, 40, 43)
	165	(22, 22, 22, 22, 32, 32, 32, 32)	(18, 24, 21, 28, 29, 34, 34, 36)
	265	(22, 22, 22, 22, 28, 28, 28, 28)	(17, 19, 18, 21, 26, 31, 30, 31)
	365	(22, 22, 22, 22, 24, 24, 24, 24)	(14, 16, 17, 27, 22, 29, 29, 29)
loot	65	(36, 36, 36, 36, 36, 36, 36, 36)	(34, 37, 37, 42, 33, 38, 37, 40)
	165	(26, 26, 26, 26, 28, 28, 28, 28)	(23, 28, 25, 33, 27, 32, 29, 31)
	265	(22, 22, 22, 22, 26, 26, 26, 26)	(17, 24, 22, 24, 24, 28, 28, 27)
	365	(22, 22, 22, 22, 22, 22, 22, 22)	(19, 18, 18, 23, 22, 26, 24, 26)
longdress	180	(30, 30, 30, 30, 38, 38, 38, 38)	(26, 32, 34, 32, 36, 37, 38, 39)
	360	(24, 24, 24, 24, 32, 32, 32, 32)	(23, 22, 26, 24, 32, 33, 34, 34)
	640	(22, 22, 22, 22, 28, 28, 28, 28)	(17, 17, 20, 19, 29, 29, 31, 31)
	840	(22, 22, 22, 22, 26, 26, 26, 26)	(16, 16, 18, 14, 28, 29, 28, 28)

5 Conclusion

In this deliverable, we proposed two DE-based solutions to optimize the QPs for the rate-distortion optimization problem for V-PCC. The first solution applies DE to the analytical rate and distortion models, while the second one applies it to the actual rate and distortion functions. In Deliverable D4, we will provide detailed experimental results for the proposed algorithms and compare them to the state of the art.

6 Appendix A: Quantization

Relationship between QP and quantization step (Q_{step})

QP	Q_{step}	QP	Q_{step}	QP	Q_{step}	QP	Q_{step}	QP	Q_{step}	QP	Q_{step}	QP	Q_{step}
0	0.625	**8**	1.625	**16**	4	**24**	10	**32**	26	**40**	64	**48**	160
1	0.6875	**9**	1.75	**17**	4.5	**25**	11	**33**	28	**41**	72	**49**	176
2	0.8125	**10**	2	**18**	5	**26**	13	**34**	32	**42**	80	**50**	208
3	0.875	**11**	2.25	**19**	5.5	**27**	14	**35**	36	**43**	88	**51**	224
4	1	**12**	2.5	**20**	6.5	**28**	16	**36**	40	**44**	104	NA	
5	1.125	**13**	2.75	**21**	7	**29**	18	**37**	44	**45**	112		
6	1.25	**14**	3.25	**22**	8	**30**	20	**38**	52	**46**	128		
7	1.375	**15**	3.5	**23**	9	**31**	22	**39**	56	**47**	144		

7 References

1. MPEG-3DG. V-PCC Codec Description. ISO/IEC JTC1/SC29/WG7 N00012, (2020).
2. MPEG-3DG. G-PCC Codec Description v9. ISO/IEC JTC1/SC29/WG7 N0011, (2020).
3. Sullivan, G. J., Ohm, J., Han, W., and Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology 22(12) 1649-1668 (2012).
4. V-PCC Test Model v12, ISO/IEC JTC 1/SC 29/WG 7 N00006.
5. Yuan, H. and Hamzaoui, R. Report on Analytical Models. Deliverable D2 of Optimized Dynamic Point Cloud Compression (OPT-PCC) Project. April 2021. https://dora.dmu.ac.uk/handle/2086/20948.
6. Price, K., Storn, R. M., and Lampinen, J. A. Differential Evolution: A Practical Approach to Global Optimization. Springer. 2005.
7. Liu, Q., Yuan, H., Hou, J., Liu, H., and Hamzaoui, R. Model-based encoding parameter optimization for 3D point cloud compression. In: Proc. APSIPA Annual Summit and Conference, pp. 1981-1986, Honolulu, (2018).
8. Liu, Q., Yuan, H., Hou, J., Hamzaoui, R., and Su, H. Model-based joint bit allocation between geometry and color for video-based 3D point cloud compression. IEEE Transactions on Multimedia, DOI: 10.1109/TMM.2020.3023294.
9. Liu, Q., Yuan, H., Hamzaoui, R., and Su, H. Coarse to fine rate control for region-based 3D point cloud compression. In: Proc. IEEE ICME Workshops, London, (2020).
10. Point Cloud Library (PCL), http://pointclouds.org/
11. Lampinen, J. and Zelinka, I. On the stagnation of the differential evolution algorithm. In: Proc. MENDEL’00 6th International Mendel Conference on Soft Computing, Brno, pp. 76-83, 2000.
12. Neri, F. and Tirronen, V. Recent advances in differential evolution: a survey and experimental analysis. Artif Intell Rev 33, 61–106 (2010).
13. Das, S., Konar, A., and Chakraborty, U.K. Two improved differential evolution schemes for faster global search, In: Proc. 7th Annual Conf. on Genetic and Evolutionary Computation GECCO’05, New York, 2005.

