Global Rate-distortion Optimization of Video-based Point Cloud Compression with Differential Evolution*

Hui Yuan School of Engineering and Sustainable Development
De Montfort University
Leicester, UK
School of Control Science and Engineering
Shandong University
Ji’nan, China
hui.yuan@dmu.ac.uk; huiyuan@sdu.edu.cn

Raouf Hamzaoui School of Engineering and Sustainable Development
De Montfort University
Leicester, UK
rhamzaoui@dmu.ac.uk

Ferrante Neri School of Computer Science
University of Nottingham
Nottingham, UK
ferrante.neri@nottingham.ac.uk

Shengxiang Yang School of Computer Science and Informatics
De Montfort University
Leicester, UK
syang@dmu.ac.uk

Tingting Wang School of Control Science and Engineering
Shandong University
Ji’nan, China
wangtingtingsdu@foxmail.com

Abstract

In video-based point cloud compression (V-PCC), one geometry video and one color video are generated from a dynamic point cloud. Then, the two videos are compressed independently using a state-of-the-art video coder. In the Moving Picture Experts Group (MPEG) V-PCC test model, the quantization parameters for a given group of frames are constrained according to a fixed offset rule. For example, for the low-delay configuration, the difference between the quantization parameters of the first frame and the quantization parameters of the following frames in the same group is zero by default. We show that the rate-distortion performance of the V-PCC test model can be improved by lifting this constraint and considering the rate-distortion optimization problem as a multi-variable constrained combinatorial optimization problem where the variables are the quantization parameters of all frames. To solve the optimization problem, we use a variant of the differential evolution algorithm. Experimental results for the low-delay configuration show that our method can achieve a Bjøntegaard delta bitrate of up to -43.04% and more accurate rate control (average bitrate error to the target bitrate of 0.45% vs. 10.75%) compared to the state-of-the-art method, which optimizes the rate-distortion performance subject to the test model default offset rule. We also show that our optimization strategy can be used to improve the rate-distortion performance of two-dimensional video coders.

Keywords: Point cloud compression, rate-distortion optimization, rate control, video coding, differential evolution.

1 Introduction

Point clouds are representations of three-dimensional (3D) objects in the form of a sample of points on their surface. Point clouds have received increased attention as they provide a better immersive experience, such as 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.
Figure 1: V-PCC video sequences. Left: geometry video. Right: color video.

are typically represented by the geometry and color (texture) of millions of 3D points. In 2017, the Moving Picture Experts Group (MPEG) launched a call for proposals for point cloud compression [1]. 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). In this paper, we focus on V-PCC [2], which has reached the Final Draft International Standard status. V-PCC can support many applications, including six degrees of freedom immersive media, virtual reality, augmented reality, and immersive real-time communication. It converts a dynamic point cloud into one geometry video and one color video (Fig. 1) and then uses a two-dimensional (2D) video coder (e.g., H.265/HEVC [3]) to compress the two video sequences independently. In the video coding step, compression is achieved with quantization, which is determined by a quantization step or, equivalently, a quantization parameter (QP) (Table 1). In the latest MPEG V-PCC test model [4], the QPs for the geometry and colour information are not optimized. For a given group of frames, one chooses the QPs for the first frame, and the QPs for the following frames in the same group are set according to a fixed difference (e.g., zero as the default value in the low-delay configuration). This approach has two major limitations. First, given a target bitrate, it is unclear how the QPs for the first frame should be chosen. Second, the imposed relationship between the QPs of the first frame and those of the following frames may harm the rate-distortion performance. To address these limitations, we propose a general rate-distortion optimization framework for V-PCC, where we allow the QPs to take any value in the admissible set. Then, we use a variant of the differential evolution (DE) algorithm [5] to optimize the selection of the QPs for each frame in a group of frames. DE is an evolutionary algorithm that is easy to implement and that has been successfully applied to various global optimization problems in science and engineering [6]. Unlike many traditional optimization techniques, it can be used when the objective function is nondifferentiable and even when it is not given in closed form. While DE was initially introduced for continuous optimization problems, it can also be adapted [6] to combinatorial optimization problems.

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

\[
\min_{Q_g, Q_c} [D_g(Q_g, Q_c), D_c(Q_g, Q_c)]
\]

subject to \( R(Q_g, Q_c) \leq R_T \)

where \( Q_g = (Q_{g,1}, Q_{g,2}, \ldots, Q_{g,N}) \), \( Q_c = (Q_{c,1}, Q_{c,2}, \ldots, Q_{c,N}) \), \( D_g(Q_g, Q_c) \) is the geometry distortion, \( D_c(Q_g, Q_c) \) is the color distortion, and \( R_g(Q_g, Q_c) \) is the number of bits allocated to the geometry and color information. Then we scalarize problem (1) as

\[
\min_{Q_g, Q_c} \omega D_g(Q_g, Q_c) + (1 - \omega) D_c(Q_g, Q_c)
\]

subject to \( R(Q_g, Q_c) \leq R_T \)

where \( \omega \) is a weighting factor that sets the relative importance of the geometry and color distortions. As the number of candidates is \( M^{2N} \), finding an optimal solution with full search is not feasible when \( M \) or \( N \) is large.

2 Related Work

Rate-distortion optimization for V-PCC has not been studied sufficiently. In [7], statistical analysis is used to derive mathematical models for the rate and distortion under the offset constraint \( QP_{g,i} = QP_{g,1} \) and \( QP_{c,i} = QP_{c,1} \) for \( i = 2, \ldots, N \), where \( QP_{g,i} \) and \( QP_{c,i} \) are the geometry QP and color QP for the ith frame in a group of N frames, respectively. Specifically, the geometry distortion \( D_g \) is modeled as \( \alpha_g Q_{g,1} + \beta_g \), the color distortion \( D_c \) is modeled as \( \alpha_{gc} Q_{c,1} + \alpha_{cc} Q_{c,1} + \beta_c \), the geometry bitrate is modeled as \( \gamma_g Q_{g,1}^{\delta} \), and the color bitrate is modeled as \( \gamma_c Q_{c,1}^{\delta} \), where \( \alpha_g, \beta_g, \alpha_{gc}, \alpha_{cc}, \beta_c, \gamma_g, \beta_g, \gamma_c, \theta_g, \gamma_c, \theta_c \) are parameters for the models. Then, problem (2) is written as

\[
\min_{Q_{g,1}, Q_{c,1}} a Q_{g,1} + b Q_{c,1} + c
\]

subject to \( \gamma_g Q_{g,1}^{\delta} + \gamma_c Q_{c,1}^{\delta} \leq R_T \) (3)
and solved using an interior point method. Experimental results show that the method has similar rate-distortion performance to exhaustive search (for the same offset constraint) but has a much lower time complexity. In [8], the method in [7] is first used to allocate the total bitrate between the geometry and color information. Then, a frame is partitioned into seven regions, and analytical models for the rate and distortion of each region are derived. Next, the geometry quantization steps for all regions are determined by minimizing the total geometry distortion subject to the allocated geometry bitrate. Finally, given the optimal geometry quantization steps, the color quantization steps for the regions are determined by minimizing the total color distortion subject to the allocated color bitrate. Experimental results for static point clouds show that the method allows more accurate rate control than the method in [7] but has lower rate-distortion performance. While the two methods are computationally efficient, their rate-distortion performance is limited by the fixed offset between the QPs of the first frame and those of the following ones. In [9], the offset constraint is lifted and DE is used to solve problem (2), where analytical models are used for the geometry and color rates and distortions. However, as the proposed analytical models lack accuracy, the rate-distortion performance is lower than that of [7].

The problem of finding the globally optimal QPs for a group of frames has been intensively studied for 2D video. By assuming a monotonicity condition, Ramchandran, Ortega, and Vetterli [10] show how an optimal solution can be found with a tree-based algorithm. However, the time complexity of the algorithm is too high, making it impractical [11]. A similar solution based on the Viterbi algorithm was proposed in [12]. However, to reduce the time complexity, a monotonicity assumption and a node clustering assumption that are not necessarily true are made. In [11], an analytical convex rate-distortion model that takes into account the dependencies between the frames is developed for HEVC. Then, a primal-dual algorithm is used to determine the optimal rates for each frame. Because the model is only an approximation of the operational rate-distortion function, the solution is not guaranteed to be optimal. Moreover, the optimization algorithm considers the problem as continuous, while the QPs are discrete. In [13], [14], [15], and [16], frames within a group of pictures are allocated different QPs according to their hierarchical layer. This approach, known as quantization parameter cascading, is motivated by the fact that frames in a lower layer have a greater impact on the overall rate-distortion performance than frames in a higher layer. However, this approach is suitable for hierarchical prediction structures only.

We conclude this section by noting that an extension of the existing 2D video rate-distortion optimization methods to V-PCC would be difficult.

### 3 Proposed optimization method

To solve the optimization problem (2), we propose to use a variant of DE. 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 \( \mu \) 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. In the following, we explain how we applied DE to the rate-distortion optimization problem (2). First, we rewrite the problem as

\[
\min_{x} D(x) = \omega D_g(x) + (1 - \omega) D_c(x)
\]

subject to \( R(x) \leq R_T \)

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

#### Parameter setting
Choose a population size \( NP \), an interval \( I \) for the scaling factor \( \mu \), and a number of iterations \( n \).

#### Initialization
Build a population of \( NP \) vectors \( X^{(1)}, \ldots, X^{(NP)} \) by randomly choosing each component \( X_{i,j}^{(k)} \), \( i = 1, \ldots, 2N; j = 1, \ldots, NP \), in the set \( \{0, 1, \ldots, M - 1\} \).

for \( k = 1 \) to \( n \)

if \( k < \frac{n}{2} \) then

Set the crossover rate \( CR \) to 0.9

else

\[ CR = 0.1 \]

end if

for \( j = 1 \) to \( NP \) do

...
Table 2: Quantization parameters ($QP_{g,1}, QP_{g,2}, QP_{g,3}, QP_{g,4}, QP_{c,1}, QP_{c,2}, QP_{c,3}, QP_{c,4}$) with the proposed method and the method in [7].

| Point cloud | Target bitrate (kbppm) | 7 | Proposed |
|-------------|------------------------|---|----------|
| Soldier     | 65                     | (36, 36, 36, 38, 38, 38, 38) | (34, 42, 40, 44, 35, 43, 42, 44) |
|             | 165                    | (28, 28, 28, 32, 32, 32, 32) | (24, 30, 31, 33, 29, 37, 33, 37) |
|             | 265                    | (4, 24, 24, 28, 28, 28, 28)  | (19, 28, 24, 27, 26, 33, 30, 34) |
|             | 365                    | (2, 22, 22, 26, 26, 26, 26)  | (18, 22, 20, 21, 25, 30, 27, 30) |
| Queen       | 65                     | (36, 36, 36, 40, 40, 40, 40) | (25, 36, 27, 33, 36, 40, 40, 43) |
|             | 165                    | (22, 22, 22, 32, 32, 32, 32) | (18, 24, 21, 28, 29, 34, 34, 36) |
|             | 265                    | (22, 22, 22, 28, 28, 28, 28) | (17, 19, 18, 21, 26, 31, 30, 31) |
|             | 365                    | (22, 22, 22, 24, 24, 24, 24) | (14, 16, 17, 27, 22, 29, 29, 29) |
| Loot        | 65                     | (36, 36, 36, 36, 36, 36, 36) | (34, 37, 34, 32, 38, 37, 40)    |
|             | 165                    | (26, 26, 26, 28, 28, 28, 28) | (23, 28, 25, 33, 27, 32, 29, 31) |
|             | 265                    | (22, 22, 22, 26, 26, 26, 26) | (17, 24, 22, 24, 28, 28, 27)    |
|             | 365                    | (22, 22, 22, 22, 22, 22, 22) | (19, 18, 18, 23, 22, 26, 24, 26) |
| Longdress   | 180                    | (30, 30, 30, 38, 38, 38, 38) | (26, 32, 34, 32, 36, 37, 38, 39) |
|             | 360                    | (24, 24, 24, 32, 32, 32, 32) | (23, 22, 26, 24, 33, 34, 34)    |
|             | 640                    | (22, 22, 22, 28, 28, 28, 28) | (17, 17, 20, 19, 29, 29, 31, 31) |
|             | 840                    | (22, 22, 22, 26, 26, 26, 26) | (16, 16, 18, 14, 28, 29, 28, 28) |

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

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

Step 3: Build a candidate vector $Y^{(j)}$ 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 $\mu$ randomly in $I$. If $r_i \leq CR$ or $i = r$, then set $Y_i^{(j)} = \text{round}(A_i + \mu(B_i - C_i))$; otherwise, set $Y_i^{(j)} = X_i^{(j)}$. Here, round means rounding to the nearest integer in the set $\{0, 1, \ldots, M - 1\}$.

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

end for

for $j = 1$ to $NP$ do

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

end for

end for

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

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 [17], 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 [18]. 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 [19], we select the scaling factor $\mu$ randomly to retain population diversity as the search progresses. Our experiments show that this randomization is beneficial. Finally, we use rounding inside the iterations to make the algorithm suitable for the combinatorial optimization problem [4].

4 Experimental results

In this section, we apply our optimization technique to V-PCC (Section 4.1) and H.264/AVC (Section 4.2). To assess the performance of our technique, we compute the Bjontegaard delta (BD) rate and BD distortion [20]. We also use the bitrate error (BE) as a measure of the bit allocation accuracy. Here, $BE = \frac{|R_a - R_T|}{R_T}$, where $R_a$ is the bitrate of the method and $R_T$ is the target bitrate. For V-PCC, the bitrates are expressed in kilobits per million points (kbppm). The source code of our method and the data files are publicly available from [21].

4.1 Rate-distortion optimization of V-PCC

We applied the proposed method to the V-PCC test model category 2 Version 12 [1], which relies on the High Efficiency Video Coding Test Model Version 16.20 to compress the geometry and color videos. We considered four dynamic point clouds (Soldier, Queen, Loot, and Longdress) and encoded their first four frames (i.e., $N = 4$) using the low-delay configuration with group of pictures (GOP) structure IPPP. In the DE algorithm, the size of the population $NP$ was 50, the number of iterations $n$ was 75, and the range $I$ of the scaling factor was the interval [0.1, 0.9]. In the initialization step, a vector was included only if it satisfied the rate constraint, where the rate was computed according to the analytical model in [9]. For the geometry and color distortions, we used the symmetric point-to-point metrics [24] based on the root mean squared error. For the color distortion, we considered only the luminance component. The weighting factor $\omega$ was set to 1/2. Table 3 compares the solutions found by our algorithm to those found by the state-of-the-art algorithm in [7]. Table 3 and Fig. 2 compare the rate-distortion performance...
Table 3: Rate-distortion performance and bitrate accuracy. The bitrates are expressed in kbpmp. The PSNRs are expressed in dB.

| Point cloud | Target bitrate | [7] | Proposed |
|-------------|----------------|-----|-----------|
|             | Bitrate | Distortion | PSNR_G | PSNR_C | BE  | Bitrate | Distortion | PSNR_G | PSNR_C | BE  |
| Soldier     | 65      | 68.60      | 27.34  | 59.13   | 30.85 | 5.54%  | 64.82     | 24.75  | 58.62   | 31.31 | 0.28% |
|             | 165     | 163.37     | 15.14  | 62.97   | 33.40 | 0.99%  | 164.90    | 13.78  | 63.14   | 33.81 | 0.06% |
|             | 265     | 296.15     | 10.91  | 65.10   | 35.41 | 13.58% | 264.47    | 10.93  | 64.86   | 34.80 | 0.20% |
|             | 365     | 414.56     | 9.51   | 65.50   | 35.41 | 13.58% | 364.96    | 9.55   | 65.77   | 35.38 | 0.01% |
| Queen       | 65      | 59.87      | 24.36  | 63.05   | 31.30 | 7.90%  | 64.98     | 21.96  | 63.57   | 31.75 | 0.03% |
|             | 165     | 162.14     | 15.29  | 65.75   | 33.32 | 1.73%  | 163.26    | 14.41  | 65.93   | 33.81 | 1.06% |
|             | 265     | 254.58     | 13.34  | 65.75   | 33.92 | 3.93%  | 261.94    | 12.37  | 66.55   | 34.24 | 1.15% |
|             | 365     | 404.43     | 10.91  | 65.75   | 34.33 | 10.80% | 359.43    | 11.43  | 66.76   | 34.58 | 1.52% |
| Loot        | 65      | 62.26      | 12.60  | 59.72   | 34.31 | 4.22%  | 64.91     | 11.32  | 59.22   | 34.83 | 0.13% |
|             | 165     | 190.11     | 5.88   | 64.27   | 37.58 | 15.22% | 164.85    | 6.17   | 63.98   | 37.37 | 0.09% |
|             | 265     | 265.31     | 4.99   | 65.51   | 38.27 | 12%    | 262.71    | 4.30   | 66.26   | 38.91 | 0.07% |
|             | 365     | 458.86     | 4.14   | 65.51   | 39.11 | 25.72% | 359.43    | 4.30   | 66.26   | 38.91 | 0.07% |
| Longdress   | 180     | 167.65     | 47.05  | 62.61   | 28.42 | 6.86%  | 179.20    | 44.20  | 62.00   | 28.69 | 0.45% |
|             | 360     | 784.40     | 29.52  | 64.62   | 30.45 | 17.85% | 359.05    | 31.37  | 64.76   | 30.18 | 0.26% |
|             | 640     | 1034.67    | 22.79  | 65.29   | 31.32 | 22.56% | 639.01    | 25.07  | 66.23   | 31.15 | 0.15% |
|             | 840     | 1384.67    | 15.85  | 65.29   | 31.57 | 23.18% | 637.97    | 22.83  | 66.66   | 31.56 | 0.24% |

Average 10.75%

Table 4: Time complexity comparison for \( n = 75 \) and \( NP = 50 \). The table shows the number of times the point cloud must be encoded.

| Method     | Initialization | Optimization |
|------------|----------------|--------------|
| [7]        | 3              | 0            |
| Proposed   | 0              | 76 x 50      |
| Full search| 0              | 52^8         |

Figure 2: Rate-distortion performance. From left to right: Soldier, Queen, Loot, Longdress.

of the proposed method to that of the method in [7]. The table provides for various target bitrates, the actual bitrate, the distortion, the peak signal-to-noise ratio (PSNR) for the geometry information (PSNR_G), the PSNR for the color information (PSNR_C), and the BE. The PSNR was computed as in [24]. Fig. 2 compares the rate-distortion curves. The results show that our method outperforms the method in [7] in terms of rate-distortion performance and bitrate accuracy. For example, the BD-rate was up to -43.04% and the highest BE was only 1.52%, while it reached 25.72% for the method in [7]. Note that the method in [7] was shown to provide results comparable to exhaustive search subject to the test model offset constraint \( QP_{g,i} = QP_{g,1}, QP_{c,i} = QP_{c,1}, i = 2, \ldots, N \). However, since the method in [7] uses mathematical models to solve the optimization problem while the proposed method uses the actual distortion and rate functions, this improvement in rate-distortion performance comes at the cost of increased computation time (Table 4). Note that the number of times our method encodes the point cloud can be reduced by saving the values of the distortions and rates in Step 4 of the DE algorithm to avoid recomputing them if the same vector is considered again. For the method in [7], the optimization process is very fast but the point cloud must be encoded three times in a pre-processing step to compute the parameters of the rate and distortion models.

4.2 Rate-distortion optimization of H.264/AVC

To show that our DE-based global optimization technique is general, we use it in this section to improve the rate-distortion performance of the x264 [25] implementation of the H.264/AVC video coding standard. For the experiments, we used the RaceHorses and BQSquare videos [26]. Both have a resolution of 416 × 240. RaceHorses has a frame rate of 30 frames per second (fps), while BQSquare’s frame rate is 60 fps. We encoded the first sixteen frames (\( N = 16 \)) using GOP structure IPP···P. For x264, we used the –bitrate rate control option, which leads to variable QP values within and across frames. We give results for both the 1-pass and the 2-pass mode. We set the other encoder settings to the default values. The settings for DE were as in the previous section, except for \( n \), which was equal to 500. Table 5 shows the results. Our optimization technique, which computes one QP for each frame, increased the average PSNR of the luminance...
component by 0.42 (resp. 0.12) dB in terms of BD-PSNR and decreased the average bitrate by 6.96% (resp. 9.48%) in terms of BD-rate compared with the 1-pass (resp. 2-pass) mode of the x264 bitrate rate control. Moreover, the average BE was only 0.16%, compared to 20.17% (resp. 6.23%) for the 1-pass (resp. 2-pass) mode.

5 Conclusion

We proposed a method to optimize the rate-distortion performance of V-PCC. Our method formulates the rate-distortion problem as a constrained combinatorial optimization problem where the optimization variables are the quantization step sizes for the compression of the geometry and color videos. A solution to the problem is computed with a variant of the DE algorithm. Experimental results show that the method outperforms the state-of-the-art interior point method in [7], which imposes a fixed offset between the quantization parameters for the first frame and those for the other frames in the same group of frames. Our method not only provides more accurate rate control but can achieve a BD rate of up to -43.04% compared to the method in [7]. The performance of our method is expected to improve if we increase the number of iterations as we used only 75 iterations, although the recommended number for such a population size and problem dimension is significantly larger [27]. Since the calculation of the actual distortion and rate requires the encoding and decoding of the point cloud, our method is only suitable for applications where the point cloud is encoded offline. Our method is very general and can be used with any video coder. For example, we showed that it can improve the rate-distortion performance of the x264 implementation of H.264/AVC. In this paper, the encoding was applied to only one group of frames of the point cloud. Extending our method to more than one group of frames is straightforward and left as future work. Our optimization framework may lead to large variations of the quantization step sizes within a group of frames, which may affect the subjective quality. This can be addressed by adding appropriate constraints on the variables in problem [4]. Other future work will include trying more sophisticated variants of DE, which have proven to be effective in other applications [6].

References

[1] Call for Proposals for Point Cloud Compression, ISO/IEC JTC1/SC29/WG11 MPEG2014/N16732, Geneva, Jan. 2017.
[2] V-PCC Codec Description, ISO/IEC JTC1/SC29/WG7 N00012, 2020.
[3] G. Sullivan, J.-R. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649-1668, 2012.
[4] V-PCC Test Model v12, ISO/IEC JTC1/SC29/WG7 N00006.
[5] K. Price, R. M. Storn, and J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization. Springer. 2005.
[6] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," IEEE Trans. Evol. Comput., vol. 15, pp. 4-31, Feb. 2011.
[7] Q. Liu, H. Yuan, J. Hou, R. Hamzaoui, and H. Su, "Model-based joint bit allocation between geometry and color for video-based 3D point cloud compression," IEEE Trans. Multimedia, DOI: 10.1109/TMM.2020.3023294.
[8] Q. Liu, H. Yuan, R. Hamzaoui, and H. Su, "Coarse to fine rate control for region-based 3D point cloud compression," in: Proc. IEEE Int. Conf. on Multimedia & Expo Workshops (ICMEW), London, July 2020.
[9] H. Yuan, R. Hamzaoui, F. Neri, and S. Yang, "Model-based rate-distortion optimized video-based point cloud compression with differential evolution," in: Proc. 11th International Conference on Image and Graphics (ICIG 2021), Haikou, China, August 2021.
[10] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," IEEE Trans. Image Process., vol. 3, pp. 533-545, Sept. 1994.
[11] A. Fiengo, G. Chierchia, M. Cagnazzo, and B. Pesquet-Popescu, "Rate allocation in predictive video coding using a convex optimization framework," IEEE Trans. Image Process., vol. 26, pp. 479-489, Jan. 2017.
[12] K. Huang and H. Hang, "Consistent picture quality control strategy for dependent video coding," IEEE Trans. Image Process., vol. 18, pp. 1004-1014, May 2009.

[13] X. Li, P. Amon, A. Hutter, and A. Kaup, "Adaptive quantization parameter cascading for hierarchical video coding," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May/Jun. 2010, pp. 4197-4200.

[14] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. J. Sullivan, High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description, JCTVC-R1002, ITU-T/ISO/IEC, Jun./Jul. 2014.

[15] T. Zhao, Z. Wang and C. W. Chen, ”Adaptive quantization parameter cascading in HEVC hierarchical coding,” IEEE Trans. Image Process., vol. 25, no. 7, pp. 2997–3009, July 2016.

[16] I. Marzuki and D. Sim, ”Perceptual adaptive quantization parameter selection using deep convolutional features for HEVC encoder,” IEEE Access, vol. 8, pp. 37052-37065, 2020.

[17] J. Lampinen and I. Zelinka, “On the stagnation of the differential evolution algorithm,” in: Proc. MENDEL’00 6th International Mendel Conference on Soft Computing, Brno, pp. 76-83, 2000.

[18] F. Neri and V. Tirronen, ”Recent advances in differential evolution: a survey and experimental analysis,” Artif Intell Rev, vol. 33, pp. 61–106, 2010.

[19] S. Das, A. Konar, and U. K. Chakraborty, ”Two improved differential evolution schemes for faster global search,” in: Proc. 7th Annual Conf. on Genetic and Evolutionary Computation GECCO’05, New York, 2005.

[20] G. Bjontegaard, Calculation of Average PSNR Differences between RD Curves, ITU-T SC16/Q6, VCEG-M33, Austin, TX, Apr. 2001.

[21] H. Yuan, R. Hamzaoui, F. Neri, S. Yang, and T. Wang. (2021). Data files for IEEE MMSP 2021 paper (1.0). Zenodo. https://doi.org/10.5281/zenodo.5211174

[22] http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/pointCloud/CfP/datasets/

[23] http://mpegfs.int-evry.fr/mpegcontent/

[24] R. Mekuria, Z. Li, C. Tulvan, and P. Chou, Evaluation Criteria for PCC (Point Cloud Compression), ISO/IEC JTC1/SC29/WG11, N16332, Geneva, June, 2016.

[25] http://www.videolan.org/developers/x264.html

[26] ftp://ftp.tnt.uni-hannover.de/testsequences/

[27] P. N. Suganthan et al., Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, 2005.