


## RESEARCH ARTICLE

# Application of MPEG-Based 3D Animation Compression Algorithm in Architectural Animation Customization

XI WANG<sup>1</sup>, KUN YU<sup>2</sup>, AND YIMING JI<sup>3</sup> <sup>1</sup>School of Arts, Hubei University, Wuhan 430000, China<sup>2</sup>School of Art and Design, Wuchang University of Technology, Wuhan 430000, China<sup>3</sup>School of Journalism and Cultural Communication, Zhongnan University of Economics and Law, Wuhan 430000, China

Corresponding author: Yiming Ji (Eltongxun@163.com)


**ABSTRACT** In recent years, 3D animation technology has developed at a rapid pace and has been used in a variety of fields. In the field of architecture, the use of bespoke 3D animation techniques to simulate the exterior and interior details of a building can not only help designers to communicate their design ideas, but can also be used in marketing to give purchasers a more visual understanding of the building. However, the production process of 3D animation is limited by storage capacity and bandwidth. In order to reduce the load of 3D data on the bandwidth, it is necessary to choose an appropriate compression method to compress the 3D data. This study introduces the Motion Picture Experts Group format algorithm to optimise the traditional algorithm for compressing 3D animation in the hope of achieving better compression results. The experimental results showed that for the quantization table filling method, linear function compression worked best. For the choice of sub-block size, the compression effect of sub-block size of A is the best and the reconstruction error is lower. The compression effect is significantly improved by introducing a local linear embedding algorithm for dimensionality reduction, and the combination of a digital zero processing algorithm and a discrete cosine transform results in better compression. In comparative experiments of 3D animation compression based on the Motion picture expert group format algorithm, the Motion Picture Experts Group format algorithm outperforms other algorithms. The experiments confirm the superior compression performance of the Motion picture expert group format algorithm and show that the Motion picture expert group format algorithm has some application value in architectural animation adaptation.

**INDEX TERMS** Architectural animation customization, 3D animation, discrete cosine transform, local linear embedding, motion picture expert group format.

## I. INTRODUCTION

With the development of science and technology, 3D animation technology is used in various fields. In the field of construction, 3D animation technology is used to simulate information in construction projects and to build 3D models that represent the specific construction and survey situation in construction projects. On the basis of the three-dimensional model of the construction information, the study introduces a

dimension in the direction of time, and the use of professional analysis software makes it possible to present the situation on the construction site in the form of a three-dimensional animation. Construction workers on site can adjust the layout and progress of the project accordingly, which has a positive impact on the implementation of the construction project [1]. However, as the complexity of 3D animation models increases, the need for more efficient computation and more storage space increases, which puts a strain on the equipment load and results in compromised data transfer and display. The reason why 3D animation data requires a large amount

The associate editor coordinating the review of this manuscript and approving it for publication was Rajeeb Dey .

of storage space is that it contains a large amount of fixed-point information [2]. Therefore, the information must be compressed before the data is stored. A large number of scholars have researched the compression of 3D animation, and have also developed corresponding standards, and a large number of compression methods have emerged, however, traditional compression schemes can cause a certain degree of damage to the data, resulting in data loss [3]. The Motion picture expert group format (MPEG) is a well-known video coding standard for compressing video data to improve transmission efficiency and image quality [4]. MPEG is characterised by its efficiency, reliability, scalability and low impact on visual effects, and it is widely used in video codecs, such as network video, TV video, etc. MPEG is characterised by its high efficiency, reliability, scalability and low impact on visual effects, and is widely used in video codecs such as network video and television video, etc. MPEG is also able to improve compression efficiency while guaranteeing the integrity of the compressed data [5]. In view of this, this research is based on the MPEG algorithm for 3D animation compression, with the expectation of reducing the temporal and spatial redundancy of 3D animation frames, thus saving storage space and improving compression performance. The data is first structured using the eK-means algorithm to select the optimal quantization table value change method and the optimal number of sub-blocks. Based on this, the MPEG algorithm is then used to design the compression process, which optimizes the 3D animation compression process based on the characteristics of the three frames in I, P and B. The research is divided into four main parts. The first part is a brief overview of the research related to compression methods. The second part details the algorithms used when compressing 3D animations based on the MPEG method. The MPEG-based 3D architectural animation compression method is then optimised according to frame characteristics. The third part validates the performance of the MPEG-based compression method by conducting performance tests and comparative analysis experiments. The fourth part is a summary and outlook of the research.

## II. RELATED WORKS

MPEG, a common compression method, is well established and Tu. C and Takeuchi. E et al. found that the application of cloud data becomes increasingly difficult as the amount of data increases in autonomous driving systems. To address the application of cloud data, an MPEG-based compression method was proposed that simulates the operation of LIDAR using mapping and localisation. The results showed that this method made the allocation of reference frames more reasonable, provided new ideas for frame prediction, and improves the performance and stability of the algorithm [6]. Nam S.& Ahn W. et al. expect to improve the performance of detecting double compression based on MPEG. The study captures special points in the discrete cosine transform based on a neural network and counts the data of the discrete cosine transform on the resolution block by quantization and

encoding of the sub-block. At the same time, the study uses the vectorisation features of the quantization table as auxiliary information as a way to improve the detection performance [7]. Yang S. and Hu Y. have designed a face-based image coding framework to address the limitations of image coding. The principle is to use a compression model and a generative model to support human perception and machine vision tasks. First, image information is input, and then feature analysis and generative models are applied to build, colour-rich and tightly structured images. The edge map acts as a base layer in the machine task and the reference pixels as an enhancement layer, in this way ensuring a more realistic visual effect. Finally simulation experiments are carried out and the results validate the superiority of the model proposed in the study, providing a data base and theoretical support for the MPEG standard [8]. Akhtar A. and Gao W., et al. found that the process of MPEG-based video compression can lead to compression artefacts due to quantization errors in the encoding. The study proposes a new solution to the artefact problem. This method was able to reconstruct the quality of the video without increasing the cost of broadband. The results showed that this method was efficient and feasible to improve the quality of point clouds and avoid artefacts [9]. Ta. A and Abl. B et al. found less research related to considering human variables as a source of ordered variation in the field of quality of experience in multimedia services. A model for enhancing user visual quality based on MPEG technology was proposed. The results showed that the quality of user visual experience was improved [10].

In the area of image compression, Bing D. U. and Duan Y., et al. propose a new compression algorithm for the development and operation of web services in the Internet of Things, which can be used directly for semantic reasoning. Semantic reasoning is then combined with image compression through a collaborative image and classification framework. Image compression and classification are performed using adversarial networks in a multi-task generation process. The results showed that this algorithm delivered better visual results [11]. Nesteruk S.& Shadrin D., et al. have proposed a compression method for receiving image data in a message transmission between Antarctica and Europe. Practical application showed that this method could be transmitted efficiently over weak channels and that close to seven image compressions were accomplished in the process. A high accuracy rate was achieved when using this algorithm for classification tasks [12]. Newman E. and Kilmer M. E. proposed a new patch learning algorithm. The study uses a patch dictionary of image data for image compression and deblurring. The process of enhancing the sparsity of the tensor coefficients is first demonstrated on the basis of a residual parametric variant. Training is then performed on classified image data, using the patch dictionary to express image data of different resolutions and categories. Finally the effect of image deblurring is investigated. The results showed that the proposed method was better processed in the study compared to the traditional method [13]. Liu Z. and Meng L., et al. have

proposed a new variable speed image compression method. This method also introduces scalar quantization with random rounding. To further improve the compression performance, the residuals of the input and reconstructed images are encoded into an enhancement layer. Experimental results showed that this method outperformed common compression methods [14]. Zhang S. and Zhang M., et al. designed a new compression scheme to address the problem of low bandwidth for image transmission underwater. First, an image pre-processing template is specifically designed to match the underwater colours, and then the original image is fed into this template. A compression framework is then developed which introduces an extreme learning machine to perform component analysis, which is distinct from lossy compression methods. Finally, the compressed image is used to recover the colours of the underwater image. The results showed that this method improved the compression rate.

In summary, MPEG as a compression method has achieved better results in various fields, but there is still relatively little research related to the compression of 3D architectural animation. In view of this, this research will investigate the 3D animation compression algorithm based on MPEG and apply this in the customisation of architectural animation.

### III. MPEG-BASED COMPRESSION METHOD FOR 3D ARCHITECTURAL ANIMATION

Custom architectural animations not only express design concepts better, but also advance architectural projects efficiently. This study compresses 3D architectural animations based on the MPEG algorithm, expecting better compression of custom architectural animations. MPEG is a classical video compression method with high compression quality. First, the development of MPEG technology and discrete cosine transform is introduced, then the quantization filling method is optimised and the influence of sub-block size is described. Then the need to use the eK-means algorithm to develop the data structuring process is explained. Finally, the MPEG method is optimised using a digital null processing algorithm and a local linear embedding algorithm based on the characteristics of I, P and B frames in 3D architectural animation data.

#### A. MPEG-BASED 3D ARCHITECTURAL ANIMATION COMPRESSION METHOD FLOW

MPEG video compression techniques have been continuously optimised as the technology has evolved [16]. The first generation of MPEG compression relied heavily on the discrete cosine transform with motion compensation for its process. The fourth generation of MPEG compression upgrades the encoding of video frames to the encoding of the content of the information in the video on top of the traditional encoding techniques, as shown in Figure 1.

Figure 1 presents the upgrade content of the fourth generation compression technology, the video sequence is converted into multiple video objects, which become a rectangular box that can be manipulated and can also change

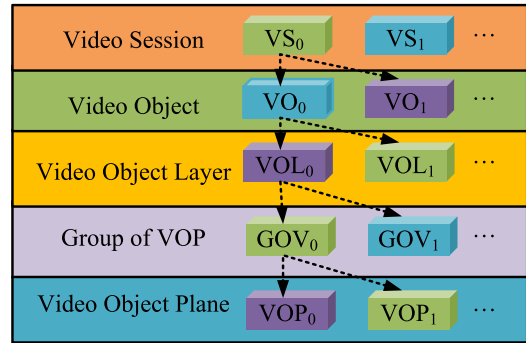


FIGURE 1. Video hierarchy.

into any object in the video scene [17]. The user can browse and search for ‘physical entities’ in the scene at any time. Video objects can usually be encoded with or without stretching. A video object contains multiple layers, each of which is a sequence of video object planes. The video object plane is a point-in-time information about the video object, containing information such as texture, shape and motion, and is the basic unit of the fourth generation compression algorithm. In addition, the fourth generation of MPEG compression technology includes the core methods of previous generations: shape coding, motion compensation and texture coding [18].

The matrix used to represent shape data in shape coding is called a bitmap and is divided into two types: binary and greyscale. A binary bitmap is represented by taking two values in a matrix that is equivalent to the boundary area of a video object. If the pixel at the corresponding position is the object of this video, the value of the element is 255. If it is not within this video object, the value of the element is 0. Before encoding, the binary bitmap is divided into 16 × 16 sub-blocks. If all elements of the binary bitmap of a video object are 0, it can be considered as a transparent binary bitmap block. If all elements of the binary bitmap of a video object are 255, then it can be considered as an opaque binary bitmap block. The representation of a greyscale bitmap has similarities to a binary bitmap, but the greyscale bitmap can take any value from 0 to 255 of the matrix elements and can also be expressed using bit bits [19]. The bit bits represent the transparency of the video pixels and are encoded using texture and data encoding. The basic operating principle of motion compensation is shown in Figure 2.

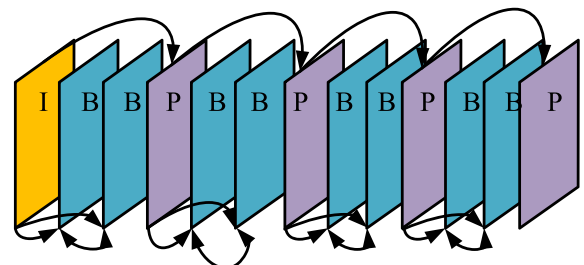


FIGURE 2. Schematic diagram of motion compensation.

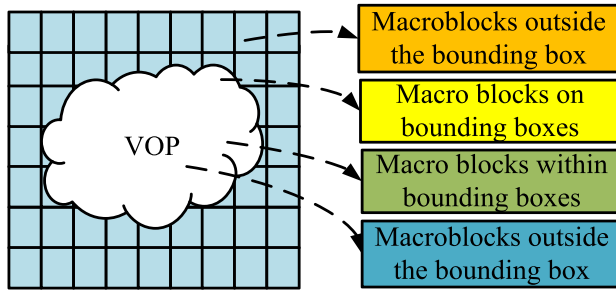


FIGURE 3. Video object planar macro block classification.

Figure 2 shows three types of video objects, in order to remove the temporal redundancy present in the video, the fourth generation MPEG compression technology uses motion compensation techniques, where the video object plane is divided into three types, mainly the three dimensions I, P and B for the encoding of intra-frame information. The division of macroblocks is used in the process of motion compensation, for which it is difficult to avoid applying to pixels outside the video object plane, which MPEG will fill [20]. In general, boundary macroblocks are filled by copying pixel points, as shown in Figure 3.

As shown in Figure 3, for macroblocks on the top and bottom, left and right, pixel points are copied and then filled vertically and horizontally, or if there are macroblocks on the top and bottom, left and right, the average of the top and bottom and the average of the left and right is used for filling. For macroblocks outside the boundary, the pixel points of the macroblocks on the adjacent boundary are copied and expanded to fill. If there is more than one adjacent macroblock, the macroblocks are filled in the priority order set by the system. If there are no adjacent macroblocks, a fill of 128 is selected. After removing the temporal redundancy present in the video, the Joint Photographic Experts Group (JPEG) video compression algorithm is used to reduce the spatial redundancy. The exact procedure is shown in Figure 4.

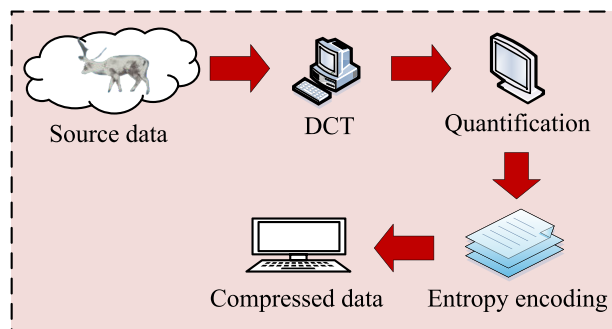


FIGURE 4. Compressing three-dimensional animation using jpeg algorithm.

Figure 4 indicates the specific compression process of the JPEG algorithm, The first step is to divide the source data into information sub blocks, with each information sub block corresponding to a sub dataset; The second step is to perform a discrete cosine transform on the data of each sub block to

obtain a matrix of discrete cosine transform coefficients; The third step is to use a quantization table to quantify the discrete cosine transform matrix; Step 4, store the quantified matrix in one-dimensional data; Step 5, entropy encode the data to obtain the final compressed 3D architectural animation. This is a complete compression process. After compression, decompression is performed using traditional methods to obtain a clear 3D architectural animation before compression.

$$F(u, v) = \frac{1}{4}C(u)V(v) \times \left[ \sum_{i=0}^7 \sum_{j=0}^7 B_k(i, j) \cos \frac{(2i+1)u\pi}{16} \right] \quad (1)$$

As shown in equation (1), where  $B_k$  is an adjacent and non-overlapping sub-block,  $B_k(i, j)$  is the pixel value of  $(i, j)$  in the image block,  $F$  is the discrete cosine transform coefficient matrix, and  $B F(u, v)$  is the frequency coefficient of  $(u, v)$  in the matrix. Let  $x$  be the independent variable in  $C(\cdot)$ ,  $D$  when  $x = 0$ , and  $D$  when  $x > 0$ . Quantize the discrete cosine transform coefficient matrix  $F$ . The quantized matrix is shown in equation (2).

$$F^Q(u, v) = IntergerRound \frac{F(u, v)}{Q(u, v)} \quad (2)$$

As shown in equation (2), where  $Q$  denotes the quantization table and  $Q(u, v)$  denotes the quantization value at the corresponding position. When performing the discrete cosine transform, the size of the sub-size can have an impact on the effectiveness of the compression. The study set the sub-block size as shown in Figure 5.

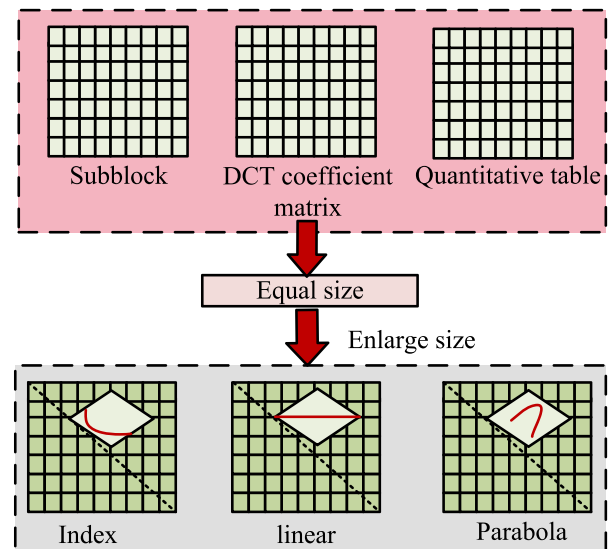


FIGURE 5. Subblock size settings.

The smaller the subsize, the greater the number and the more time it takes. And the larger the subsize, the more data is lost as a result of quantization. Therefore, a reasonable choice of subsize is needed. The upper left corner of the



discrete cosine transform coefficients stores the information before the transform. The larger the matrix size, the less space is required to store the information. As shown in Figure 5, the sub-block size is set and the quantization table is modified simultaneously. The quantization table is filled using exponential functions, parabolic functions and linear functions. The discrete cosine transform coefficients lead to integer data results after quantization, and the data is processed for reduction affecting the floating point, which can easily result in distorted 3D architectural animation results. In this study, the discrete cosine transform was changed by expanding the multiplier in the expectation that floating point preservation would be carried out to avoid information loss [21]. The improved quantization equation is shown in equation (3).

$$F^Q(u, v) = \text{IntegerRound} \frac{F(u, v) \bullet \text{Mul}}{Q(u, v)} \quad (3)$$

As shown in equation (3),  $F^Q$  denotes the result of the quantization of the discrete cosine transform. where  $\text{Mul}$  denotes the expanded multiplier. The inverse quantization equation is shown in equation (4).

$$F^R(u, v) = \frac{F^Q(u, v) \times Q(u, v)}{\text{Mul}} \quad (4)$$

As shown in equation (4),  $F^R$  represents the reduced discrete cosine transform coefficient matrix. After the discrete cosine transform, the smaller the difference between the data values within each sub-block, the more non-zero values in the original data are clustered in the upper left and zero values are clustered in the lower right. After DCT, more zeros means better compression. However, in 3D architectural animation, the ordering of vertices in each frame is not regular, resulting in differences between columns in the vertex set  $V$  that are not computable. Therefore, the sequence data in the vertex set  $V$  is rearranged in order to cluster the data of adjacent vertices before compression is performed, thus making the differences in the sequence data smaller and thus enabling the structuring of the architectural animation data. In view of this, the study introduces the eK-means algorithm, which clusters all the vertex sets together so that each class has the same size as the discrete cosine transform, resulting in a greater improvement in the performance of the compression [7]. When calculating the aggregation centre,  $V^1$  represents the result of the first frame of vertex data aggregation, using the first column of data  $V^1$  to reorder the vertex set  $V$ . eK-means the specific steps are: first select  $K$  initial aggregation points randomly in  $V^1$ , represented by  $\mu_1, \mu_2, \mu_3, \dots, \mu_K \in V^1$ ; then calculate the distance of vertex  $V_i^1$  from  $K$  in  $V^1$ , the calculation equation is shown in equation (5).

$$d = \left\| V_i^1 - \mu_j \right\|^2 \quad (5)$$

As shown in equation (5), where  $j = (1, 2, 3, \dots, K)$ , with reference to the magnitude of the distance, assigns  $V_i^1$  to the nearest centroid, forming the class  $K$ . Finally, the mean value

of each class is calculated and denoted by  $m_j$ . This is taken as the new aggregation centre and let  $\mu_j = m_j$ , based on equation (6), determine its convergence.

$$E = \sum_{j=1}^K \sum_{k \in C_i} (V_k^1 - \mu_j)^2 \quad (6)$$

As shown in equation (6), the aggregation ends when convergence reaches the maximum number of iterations, and if not the above operation is repeated. The exact flow of the algorithm is shown in Figure 6.

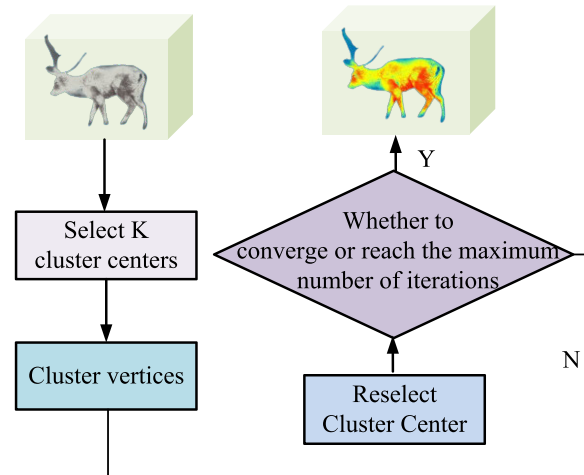


FIGURE 6. Three-dimensional animation data reordering based on eK-means clustering algorithm.

As shown in Figure 6, the results of the first frame of vertex information in the 3D architectural animation are shown after the aggregation calculation, with each of the resulting categories indicated by a different colour. Although in the eK-means algorithm both outliers and initial aggregation points can affect the results, making them unstable and prone to falling into local optima. However, the data in 3D architectural animation is inherently smooth, a property in space and time that makes the algorithm less influential on the results. Furthermore, the effect on the results only occurs when the starting point is at the boundary of the model, which is less frequent and almost non-existent. Therefore, the use of the eK-means algorithm has some applicability.

**B. IMPROVEMENTS IN COMPRESSION METHODS BASED ON MPEG 3D ANIMATION**

In the MPEG-based 3D architectural animation compression process, the video sequences are first classified into three types of frames on a time scale: I, P and B. The frames are then divided into macroblocks, which are then subjected to motion compensation, coding and discrete cosine transform operations. The compression of the three types of frames is shown in Figure 7.

Figure 7 illustrates the compression process of three types of frames in detail: I-frames, also called key frames, are

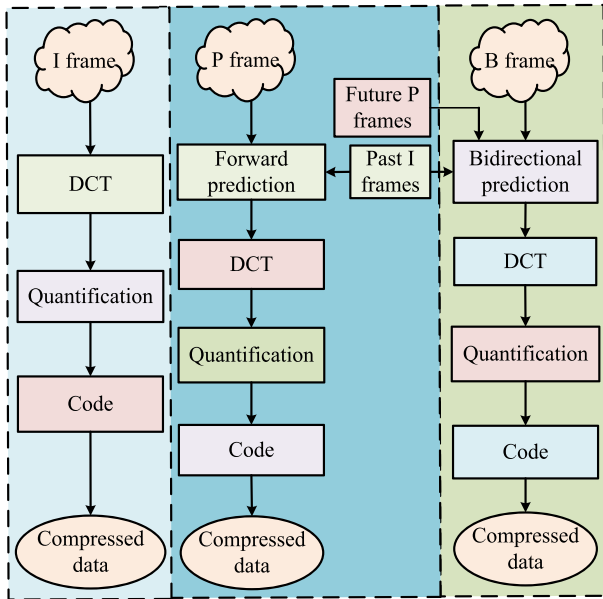


FIGURE 7. Three compression processes for frames.

compressed in an intra-frame manner, retaining all information during compression and requiring only an inverse operation during decompression, with no other additional processing. P-frames, also called forward prediction frames, mainly store the difference data between P- and I-frames, and are no longer complete after compression, requiring the use of previous frame information. B-frames, also called bi-directional prediction frames, store the information of the present frame, the difference data between the previous most recent P or I frame and the subsequent P and I frames [22]. The full compression process is shown in Figure 8.

Compression is a mechanism that reduces the size of computer files using specific algorithms. This mechanism is a very useful invention, especially for network users, as it can reduce the total number of bytes in files, allow faster transmission of files over slower Internet connections, and also reduce the amount of disk space occupied by files. In order to reduce spatial and temporal redundancy and thus improve the compression performance of the 3D architectural animation, the MPEG algorithm is used to divide the architectural animation into I, P, and B frames. These frames are then decompressed in various ways based on their unique characteristics. Figure 8 depicts the specific compression process of these three types of frames [23]. Decompression is the reverse process of compression, in which various documents, files and other objects compressed by software are restored to their original state before compression. I-frame compression can help reduce spatial redundancy, and the ability of the iframe itself to store complete information ensures the quality of the image after decompression in 3D architectural animation. The main purpose of P-frame compression is to remove spatial redundancy in conjunction with the discrete cosine transform.

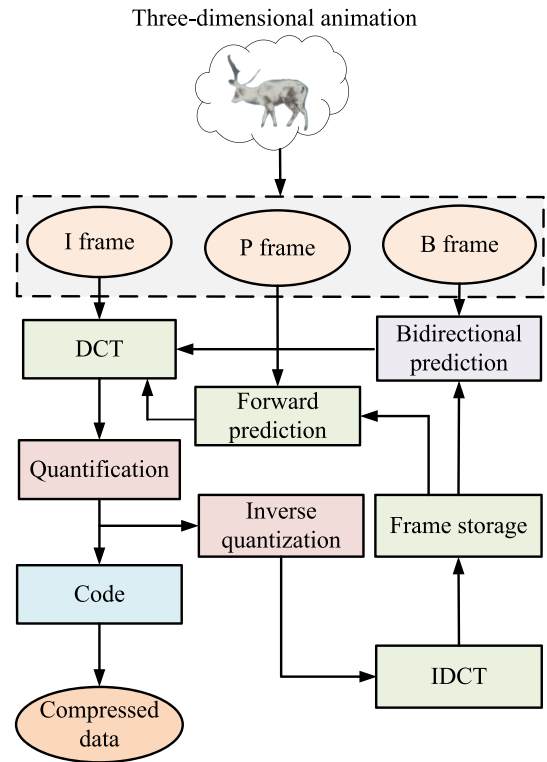


FIGURE 8. The compression process of MPEG.

The B-frame is a combination of temporal and spatial redundancy, with the B-frame compressing the corresponding data differences. Before decompression, the I- and P-frames need to be decompressed to help calculate the true B-frame information [24].

To improve the compression performance, each frame is unfolded for dimensionality reduction and then classified according to its features. The study mainly uses the local linear embedding method for dimensionality reduction, which has the main advantage of preserving the local structure of the data without using a large number of feature value extractors. In addition, the local linear embedding algorithm can effectively deal with problems such as noise and large data sets. Let the number of nearest neighbours of the sample be  $k$ . Using the weighting of the neighbours to calculate the local weights of all the vertices in the sample, the weight matrix is calculated as  $W$ , Let the number of vertices in the frame of each frame be  $m$ , then there exists  $\{x_1, x_2, x_3, \dots, x_m\}$ . The reconstructed error function is shown in equation (7).

$$E(W) = \sum_{i=1}^N \left| x_i - \sum_{j=1}^k w_{ij}x_j \right|^2 \quad (7)$$

As shown in equation (7), where  $j = (1, 2, 3, \dots, k)$ , the  $i$ th nearest neighbour is denoted by  $x_j$  and the weight between  $x_i$  and  $x_j$  vertices is denoted by  $w_{ij}$  and satisfies  $\sum_{j=1}^k w_{ij} = 1$ .

Matrixize equation (7) as shown in equation (8).

$$E(W) = \sum_{i=1}^N W_i^T (x_i - x_j)(x_i - x_j)^T W_i \quad (8)$$

The equation is shown in equation (8), where  $I$  is the  $k$ -dimensional full 1 matrix and  $\sum_{j=1}^k w_{ij} = W_i^T I = 1$ . The equation is constructed using the Lagrangian method as shown in equation (9).

$$L(W) = \sum_{i=1}^N W_i^T Q_i W_i + \lambda (W_i^T I - 1) \quad (9)$$

As shown in equation (9), where  $Q_i = (x_i - x_j)(x_i - x_j)^T$ , the derivative of  $W$  in equation (9), yields the optimal weight matrix about  $x_i$  as shown in equation (10).

$$W_i = \frac{Q_i^{-1} I}{I^T Q_i^{-1} I} \quad (10)$$

As shown in equation (10), where the value of the equation in the derivation is 0. The vertices are downsampled and the reduced coordinates are projected onto a two-dimensional plane with the error function equation shown in equation (11).

$$E(Y) = \sum_{i=1}^N \left| y_i - \sum_{j=1}^k w_{ij} y_j \right|^2 \quad (11)$$

As shown in equation (11), where the dimensionality reduction of  $x_i$  results in  $y_i$ , subject to the condition shown in equation (12).

$$\sum_{i=1}^m y_i = 0, \quad \frac{1}{m} \sum_{i=1}^m y_i y_i^T = E \quad (12)$$

As shown in equation (12), where the unit matrix of  $m \times m$  is expressed in  $E$ . The matrixed error function expression is shown in equation (13).

$$E(Y) = \text{tr}(Y(E - W)(E - W)^T Y^T) \quad (13)$$

As shown in equation (13), where  $\text{tr}(\cdot)$  is the trace function. The weight matrix  $W$  is known and the result of the dimensionality reduction is  $Y$ . If the nearest neighbour of  $x_i$  is  $x_j$ , then  $W_i = w_{ij}$ , if not, then  $W_i = 0$ . Again the Lagrangian equation is used as shown in equation (14).

$$L(Y) = \text{tr}(YMT^T + \lambda(YY^T - mE)) \quad (14)$$

As shown in equation (14), where  $M = (I - W)(I - W)^T$ , the derivative of  $Y$  is shown in equation (15).

$$MY^T = \lambda' Y^T \quad (15)$$

As shown in equation (15), where the value of the equation is 0. The prerequisite for obtaining the optimal dimensionality reduction result is to calculate the eigenvector corresponding to the smallest of the  $n$  features in the matrix  $M$ . In general, the eigenvector corresponding to the eigenvalue of  $2 \sim n + 1$  is taken as the result  $Y$ . Because the discrete cosine transform cannot completely avoid data loss in the process

Three-dimensional animation

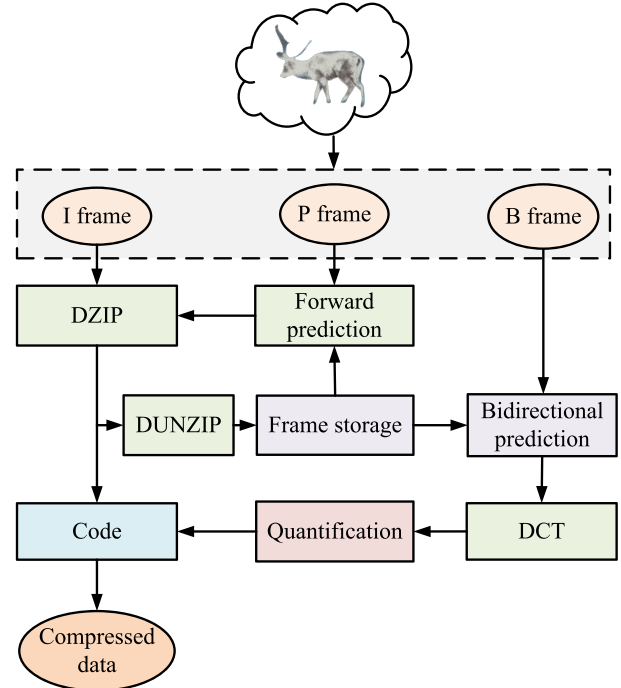


FIGURE 9. Optimized MPEG compression process.

of compression, the compression process is optimised. The optimised compression process is shown in Figure 9.

As shown in Figure 9, this method is still used to compress the B frames, which have the largest number of frames, in order to ensure the compression ratio. For the smaller number of I and P frames, the Digital Zero Processing (DZIP) algorithm is used, which uses the LZ77 algorithm for compression and then entropy coding. The main process of LZ77 is to compress all input sequences by traversing them in a sliding window, dividing the sliding window into a search buffer and a pre-buffer, and replacing long sequences with less memory consumption. The search buffer holds the part of the sequence closest to the encoded sequence, and the search buffer is empty when compression is first performed. The length of the input sequence then begins to increase backwards, and as the length increases to the maximum capacity of the buffer, the position at the window slides backwards. A front buffer stores a sequence to be encoded. After understanding the compression method for 3D architectural animation, the performance of this method needs to be evaluated. There exists  $M$  frames, and for each frame the sequence of  $N$  vertices is calculated as shown in equation (16).

$$buf = \frac{n_{encode}}{MN} \quad (16)$$

As shown in equation (16), where  $n_{encode}$  denotes the total number of bits after entropy encoding.  $buf$  is used to measure the compression ratio of the 3D architectural animation. The reconstruction error is calculated by the

equation shown in equation (17).

$$KGError = 100 \frac{\|G - \hat{G}\|_F}{\|G - E(G)\|_F} \quad (17)$$

As shown in equation (17), where the original 3D architectural animation coordinate matrix is denoted by  $G$ , the decompressed coordinate matrix is denoted by  $\hat{G}$ , the average of all frame coordinates is denoted by  $E(G)$ , and  $\|\cdot\|_F$  denotes the Frobenius parametrization. MPEG can also be used for image change detection. Luo F et al. constructed a multi-scale differential change feature fusion network based on multi-scale features, hoping to represent subtle changes between dual temporal hyperspectral images at each scale. The research method shows that MPEG technology can be used for change detection in hyperspectral images, taking into account multi-scale features including changes and invariant components, and accurately representing subtle changes between dual temporal hyperspectral images at each scale [25].

#### IV. EXPERIMENTAL EVALUATION OF MPEG-BASED COMPRESSION ALGORITHMS FOR 3D ARCHITECTURAL ANIMATION

To avoid experimental errors caused by different devices, the same device was chosen for this experiment. MATLAB R2018b was chosen to perform a series of encoding operations, and the CPU of the computer was Intel Core i7-9700 3.00GHz. The impact of the methods used in MPEG on the compression effect was first tested, and then the compression performance of MPEG methods and common compression methods were compared and analysed to verify the MPEG methods in architectural 3D building animation. The applicability of the MPEG method in architectural 3D architectural animation is verified.

##### A. PERFORMANCE TESTING OF MPEG-BASED COMPRESSION ALGORITHMS FOR 3D ARCHITECTURAL ANIMATION

To investigate the effect of the methods used in MPEG on the compression effect, four 3D animation datasets from the Architectural Animation Library, A, B, C and D, were selected to test the performance of the methods. In the quantization table fill, sub-block size  $8 \times 8$  was chosen and the effect of different fill methods on the compression ratio is shown in Figure 10.

As can be seen in Figure 10, the horizontal coordinate represents the compression ratio and the vertical coordinate represents the reconstruction error, which decreases as the compression ratio increases. For the filling with the exponential function, the compression effect is poor and the reconstruction error is large, reaching a maximum of 0.7 for the analysis of the 3D data set D. For the filling with the parabolic and linear functions, the two curves in the four data sets almost coincide, indicating that the compression ratio and reconstruction error of the two methods tend to be the

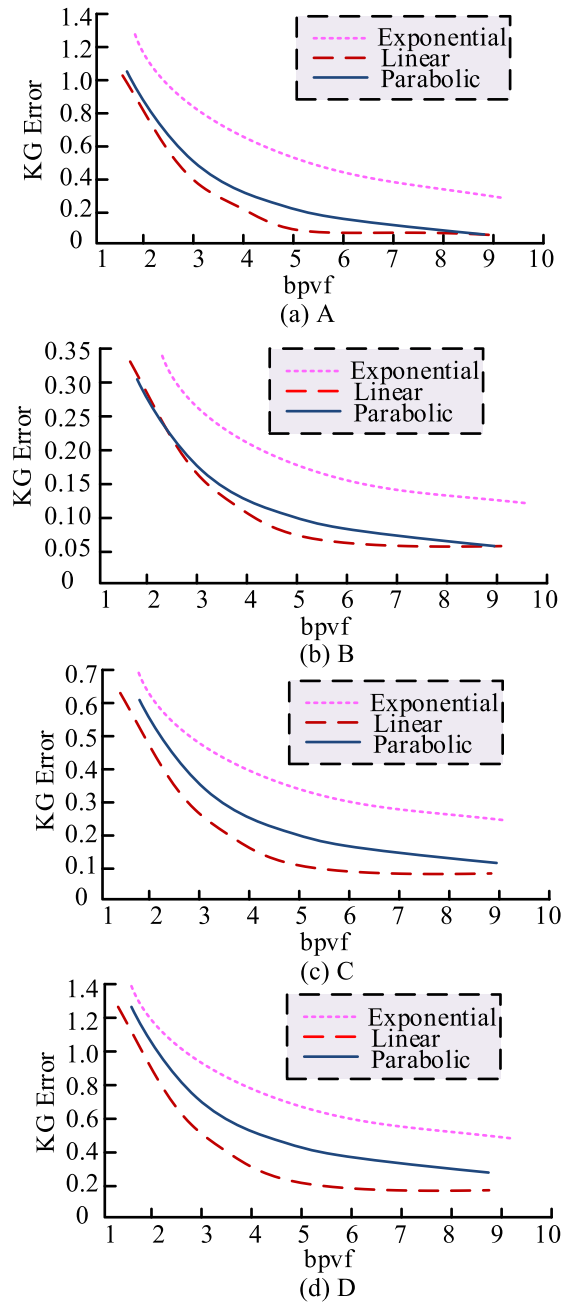
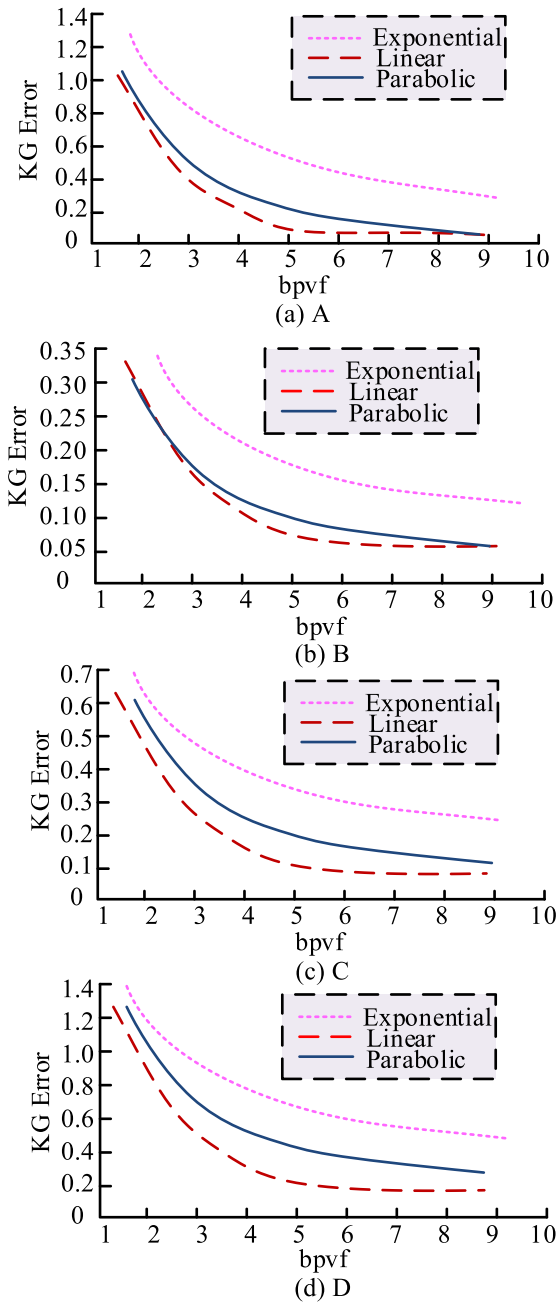


FIGURE 10. Influence of different filling methods on compressibility ratio ( $8 \times 8$ ).

same, and the average value of the lowest reconstruction error for the four data sets is 0.005. In order to select the way in which the values of the quantisation table change, sub-blocks of different sizes are selected for this study and analysis. The sub-block size of  $16 \times 16$  is chosen and the effect of different filling methods on the compressibility ratio is shown in Figure 11.

As can be seen from Figure 11, the horizontal coordinates represent the compression ratio and the vertical coordinates represent the reconstruction error, which decreases as the compression ratio increases. Filling with the exponential





**FIGURE 11.** Influence of different filling methods on compressibility ratio(16 × 16).

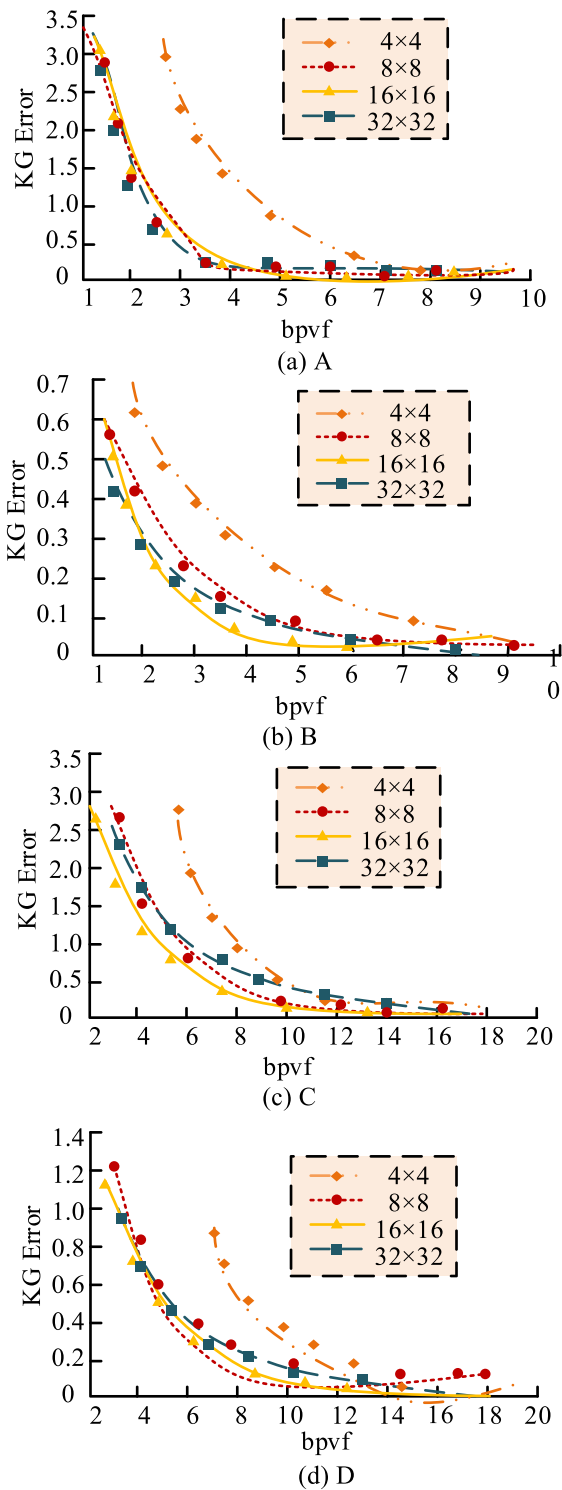
function is less effective and the reconstruction error is greater, reaching a maximum of 1.4 in the analysis of the 3D dataset D. The two curves in datasets A and B are closer together when filled with the parabolic and linear functions. The reconstruction error of the linear function is lower than that of the parabolic function in datasets C and D. Overall, the mean of the lowest reconstruction error for the four datasets of linear functions is smaller at 0.1. Combining Figure 10 and Figure 11 shows that the compression effect of the filling method using linear and parabolic functions is significantly better than the compression effect using

exponential functions, while the compression effect of linear functions is slightly better than the compression effect of parabolic functions. Therefore, the linear function was chosen as the fill method in the quantization table for this study. The size of the sub-block also has an effect on the compression effect, and the effect of different sub-block sizes on the compression performance is shown in Figure 12.

As can be seen from Figure 12, the horizontal coordinate represents the compression ratio and the vertical coordinate represents the reconstruction error, which decreases as the compression ratio increases. When the sub-block size is 4 and 8, the compression effect is poor and the reconstruction error is large. When the sub-block size is 16 and 32, the curves in the four data sets almost coincide, indicating that the compression ratio and reconstruction error tend to be the same for both sizes, and the average value of the lowest error in the four data sets is only 0.02. To select the optimal sub-block size, the compression time is used as the evaluation index, and the results of the effect of different sizes on the compression time are shown in Figure 13.

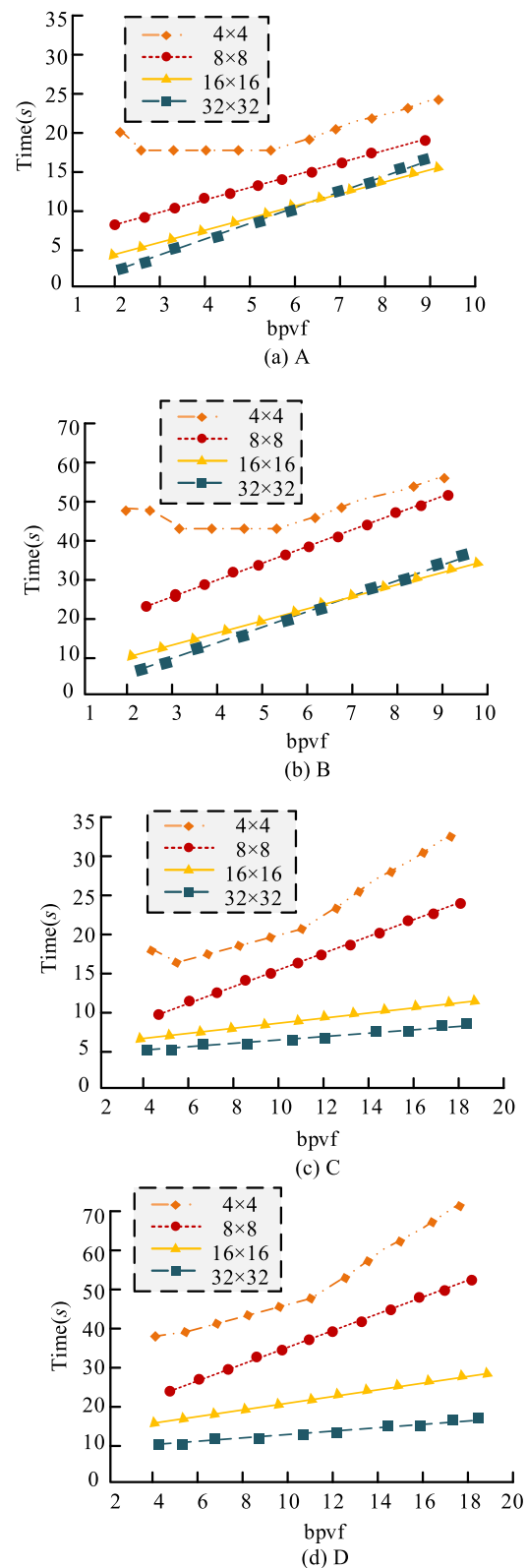
As can be seen from Figure 13, the horizontal coordinates represent the compression ratio and the vertical coordinates represent the compression time, which increases as the compression ratio increases. The compression time is greater when the sub-block size is 4 versus 8. Curves with sub-block sizes of 16 vs 32 intersect in data sets C and D and are not easily comparable. However, for data sets A and B, the compression time for a sub-block size of 32 is slightly less than the compression time for a sub-block size of 16. The average of the maximum compression time for the four datasets is 15 s when the sub-block size is 16. The average of the maximum compression time for the four data sets is 12V when the sub-block size is 32. Therefore, the sub-block size 32 × 32 is chosen for this study. To explore the applicability of the local linear embedding algorithm introduced in this study, the compression performance before and after the introduction of the local linear embedding algorithm is compared. The results of the comparison are shown in Figure 14.

As can be seen from Figure 14, the horizontal coordinate represents the compression ratio and the vertical coordinate represents the compression time, and the reconstruction error gradually decreases as the compression ratio increases. For data sets A and B, the compression effect is improved after using the local linear embedding algorithm. For data sets C and D, the compression effect is significantly improved after using the local linear embedding algorithm. This is probably because the number of frames and the number of vertices per frame are larger in datasets C and D than in datasets A and B, so the compression effect is more obvious. As 3D technology develops, the number of 3D frames and the number of vertices will increase. Therefore, using the local linear embedding algorithm for dimensionality reduction can achieve better results and has some applicability. After the dimensionality reduction process using the local linear embedding algorithm, the DZIP algorithm and the discrete



**FIGURE 12.** Influence of different sub block sizes on compression performance.

cosine transform were used for compression. In order to investigate the superiority of the compression methods used in this study, a comparison of the compression performance between the compression using the DZIP algorithm, the discrete cosine transform and the compression using only the



**FIGURE 13.** The effect of different subblock sizes on compression time.

discrete cosine transform is made and the results are shown in Figure 15.

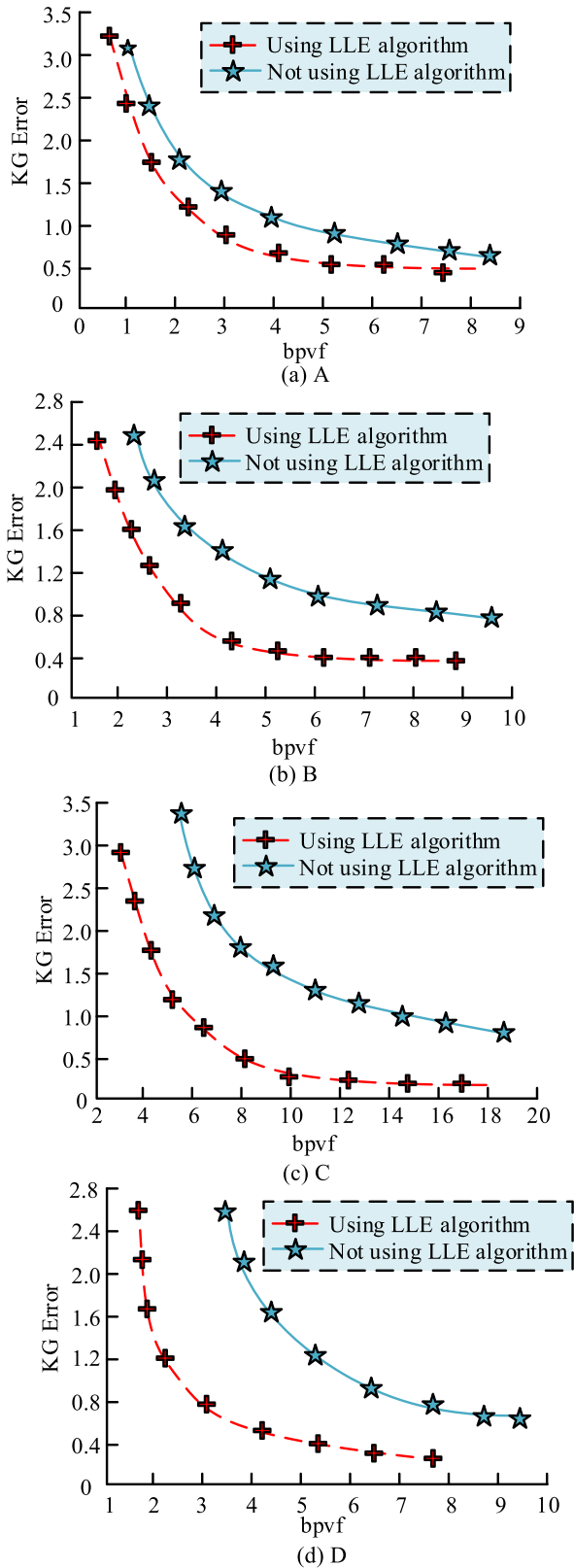


FIGURE 14. Impact of LLE algorithm on compression performance.

As can be seen from Figure 15, the horizontal coordinate represents the compression ratio and the vertical coordinate

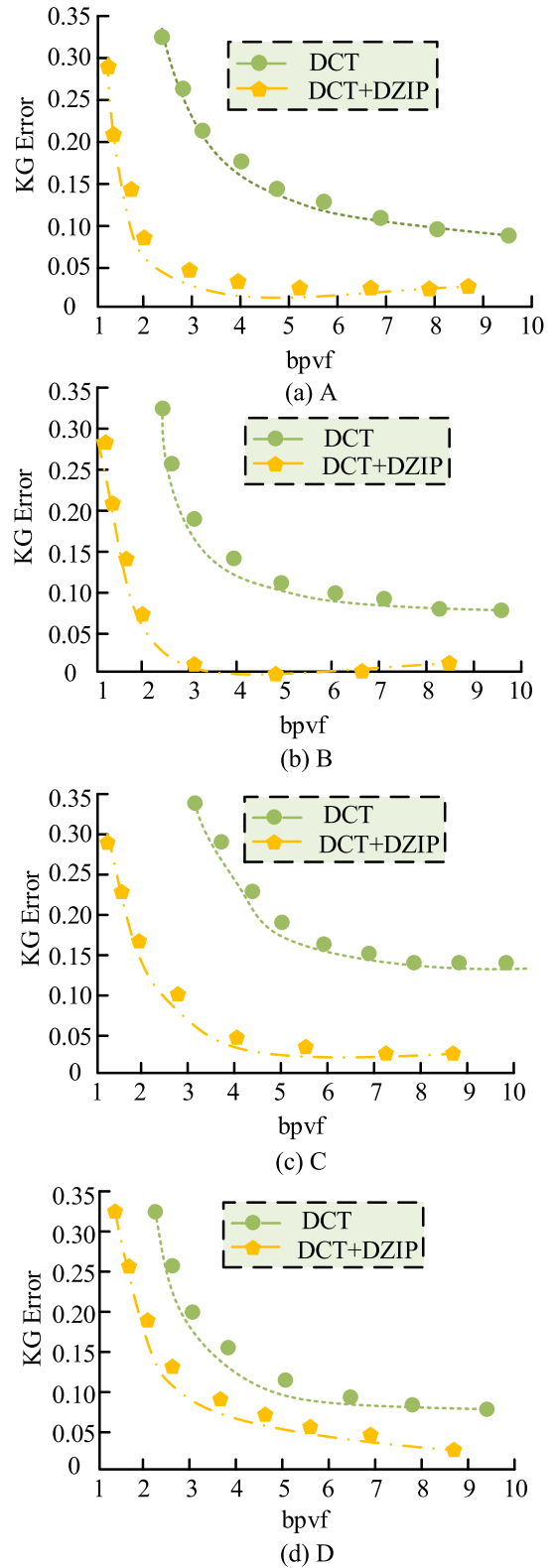


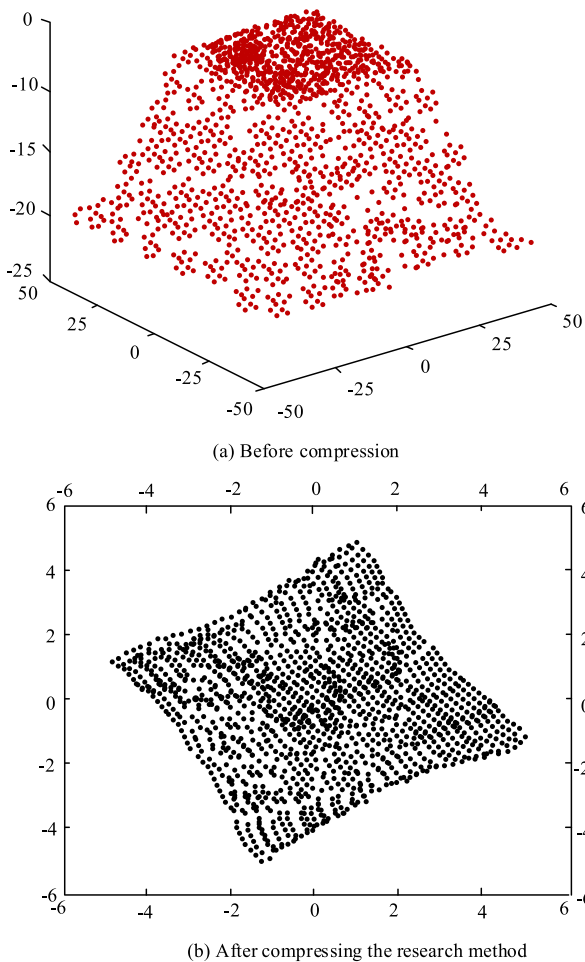
FIGURE 15. Influence of using DZIP method on compression performance.

represents the compression time, and the reconstruction error gradually decreases as the compression ratio increases. The optimization effect is significant in all four data sets, and

the compression effect is better with the DZIP algorithm and discrete cosine transform for compression, and the reconstruction error is smaller, the average value of the lowest reconstruction error in the four data sets is only 0.2. The experimental results show that the DZIP algorithm plays a greater role in optimizing the MPEG-based compression method.

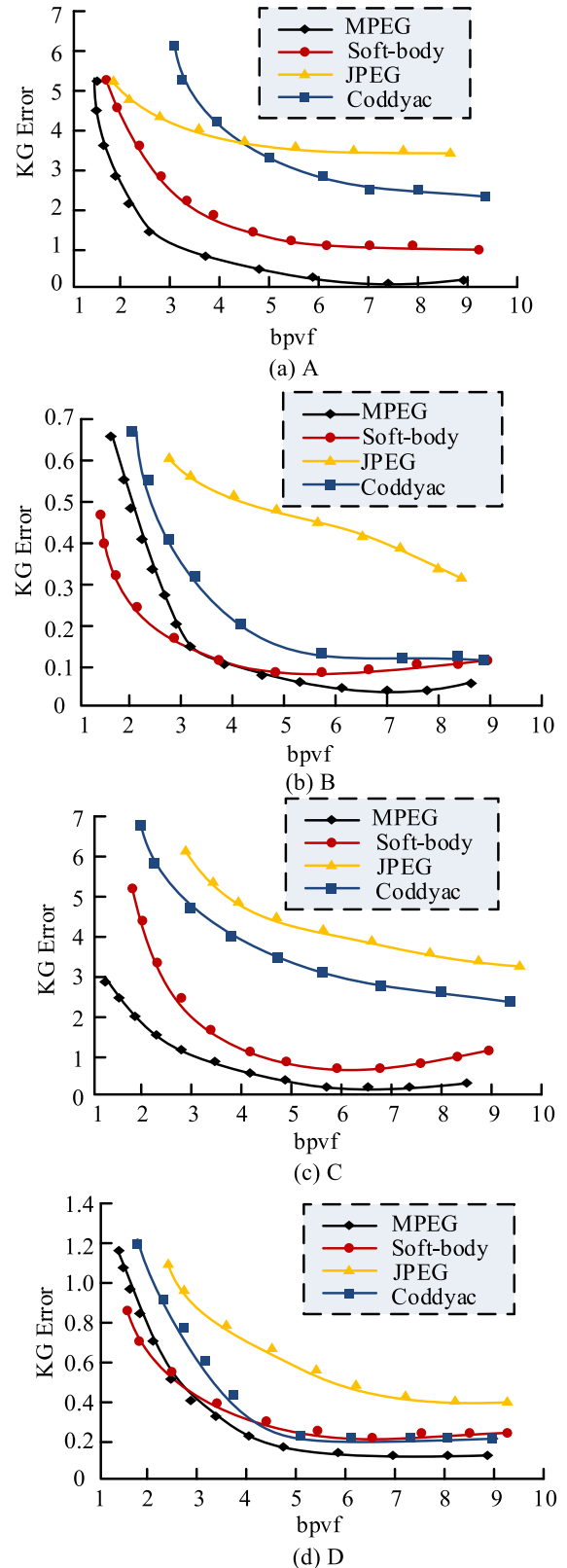
**B. COMPARATIVE ANALYSIS OF MPEG-BASED COMPRESSION ALGORITHMS FOR 3D ANIMATION**

To further verify the applicability of the research method, the research method is used to compress an image in the building database, and the eighth frame of the image is selected for compression. To make the compression effect more intuitive, it is presented in the form of visual numbers. Study algorithm, frame 8 of the image is shown in Figure Figure 16.



**FIGURE 16.** Comparison of image usage research algorithm before and after.

As shown in Figure 16 (a), this is a visual figure of the eighth frame of the image before compression, and it can be seen that its volume is large; judging from Figure 16 (b), the image volume is significantly reduced after compression, effectively saving space and achieving the compression effect. To verify the superiority of the compression algorithm



**FIGURE 17.** Comparison of compression performance of different compression methods.

based on MPEG 3D animation, the common compression methods Soft-body, Coddycac and JPEG were introduced to

compare the compression effect of the four methods, the results of which are shown in Figure 17.

As can be noticed from Figure 17, the horizontal coordinate represents the compression ratio and the vertical coordinate represents the compression time, with the reconstruction error gradually decreasing as the compression ratio increases. It is clear that JPEG is the worst compressor, followed by Coddycac. For data sets B and D, there is an intersection point between MPEG and Soft-body. Before the intersection, the reconstruction error of MPEG is greater than that of Soft-body for the same compression ratio. After the intersection, the reconstruction error of MPEG is smaller than the reconstruction error of Soft-body for the same compression ratio. MPEG has a superior compression impact and a much lower reconstruction error than Soft-body for datasets A and C. The average value of the minimum reconstruction error across the four datasets is 0.2, demonstrating that MPEG has the best effect in restoring the original data of 3D animation and has some applicability in customised architectural animation. Overall, MPEG has the best compression effect and the smallest reconstruction error.

## V. CONCLUSION

In this study, MPEG is used to compress the 3D animation in order to improve the compression performance. The linear function compresses the data best during the performance test experiments of the algorithm for the quantization table filling method, and its reconstruction error is lower than that of the parabolic and exponential functions. The average value of the lowest reconstruction error of the linear function for the four data sets is only 0.1. Among the four datasets, the average value of the lowest reconstruction error is only 0.02, and the average value of the highest compression time is only 12 s. The 3232 sub-block size has the best compression effect and the lowest reconstruction error. For datasets C and D with more frames, the compression effect is greatly improved by introducing the local linear embedding approach for dimensionality reduction. Frames I, P and B were compressed using the discrete cosine transform and the DZIP technique. The results showed that this approach was superior to relying on the discrete cosine transform alone. MPEG 3D animation compression outperformed Coddycac, JPEG and Soft-body compression in a comparative experiment, with MPEG having the lowest reconstruction error; the average value of the minimum reconstruction error for the four data sets was 0.2. The experimental findings support the improved MPEG compression performance as well as the MPEG algorithm's adaptation to 3D animation compression. Architectural 3D animation customization can help designers more effectively communicate their design thoughts, reduce design time, and increase design efficiency. The LLE technique was selected for the animation's dimensionality reduction, although it is insufficient for potential future 3D architectural animations with greater frame rates. This research still has certain flaws. In order to obtain better dimensionality reduction, future

research will integrate more accurate feature classification with additional deep learning algorithms.

## REFERENCES

- [1] G. Zhang, W. Ling, and C. Duan, "Motion damage attitude acquisition based on three-dimensional image analysis," *IEEE Sensors J.*, vol. 20, no. 20, pp. 11901–11908, Oct. 2020.
- [2] J. Liu, M. Zhang, X. Tong, and Z. Wang, "Image compression and encryption algorithm based on 2D compressive sensing and hyperchaotic system," *Multimedia Syst.*, vol. 28, no. 2, pp. 595–610, Apr. 2022.
- [3] J. Xu, J. Mou, J. Liu, and J. Hao, "The image compression-encryption algorithm based on the compression sensing and fractional-order chaotic system," *Vis. Comput.*, vol. 38, no. 5, pp. 1509–1526, May 2022.
- [4] I. Emaldi, E. Erkizia, J. R. Leiza, and J. S. Dolado, "Understanding the effect of MPEG-PCE's microstructure on the adsorption and hydration of OPC," *J. Amer. Ceram. Soc.*, vol. 106, no. 4, pp. 2567–2579, Apr. 2023.
- [5] X. Wei, M. Zhou, S. Kwong, H. Yuan, S. Wang, G. Zhu, and J. Cao, "Reinforcement learning-based QoE-oriented dynamic adaptive streaming framework," *Inf. Sci.*, vol. 569, pp. 786–803, Aug. 2021.
- [6] C. Tu, E. Takeuchi, A. Carballo, C. Miyajima, and K. Takeda, "Motion analysis and performance improved method for 3D LiDAR sensor data compression," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 243–256, Jan. 2021.
- [7] S.-H. Nam, W. Ahn, M.-J. Kwon, J. Kang, and I.-J. Yu, "DHNet: Double MPEG-4 compression detection via multiple DCT histograms," *IEEE MultimediaMag.*, vol. 29, no. 2, pp. 11–22, Apr. 2022.
- [8] S. Yang, Y. Hu, W. Yang, L.-Y. Duan, and J. Liu, "Towards coding for human and machine vision: Scalable face image coding," *IEEE Trans. Multimedia*, vol. 23, pp. 2957–2971, 2021.
- [9] A. Akhtar, W. Gao, L. Li, Z. Li, W. Jia, and S. Liu, "Video-based point cloud compression artifact removal," *IEEE Trans. Multimedia*, vol. 24, pp. 2866–2876, 2022.
- [10] T. Abar, A. B. Letaifa, and S. E. Asmi, "User behavior-ensemble learning based improving QoE fairness in HTTP adaptive streaming over SDN approach," *Adv. Comput.*, vol. 123, no. 1, pp. 245–269, Mar. 2021.
- [11] B. Du, Y. Duan, H. Zhang, X. Tao, Y. Wu, and C. Ru, "Collaborative image compression and classification with multi-task learning for visual Internet of Things," *Chin. J. Aeronaut.*, vol. 35, no. 5, pp. 390–399, May 2022.
- [12] S. Nesteruk, D. Shadrin, M. Pukalchik, A. Somov, C. Zeidler, P. Zabel, and D. Schubert, "Image compression and plants classification using machine learning in controlled-environment agriculture: Antarctic station use case," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17564–17572, Aug. 2021.
- [13] E. Newman and M. E. Kilmer, "Nonnegative tensor patch dictionary approaches for image compression and deblurring applications," *SIAM J. Imag. Sci.*, vol. 13, no. 3, pp. 1084–1112, Jan. 2020.
- [14] Z. Liu, L. Meng, Y. Tan, J. Zhang, and H. Zhang, "Image compression based on octave convolution and semantic segmentation," *Knowl.-Based Syst.*, vol. 228, no. 1, Sep. 2021, Art. no. 107254.
- [15] S. Zhang, M. Zhang, Y. Cui, X. Liu, B. He, and J. Chen, "A fast ELM-based machine compression scheme for underwater image transmission on a low-bandwidth acoustic channel," *Sensor Rev.*, vol. 39, no. 4, pp. 542–553, Jul. 2019.
- [16] A. Islam, F. Othman, N. Sakib, and H. M. H. Babu, "Prevention of shoulder-surfing attack using shifting condition with the digraph substitution rules," *Artif. Intell. Appl.*, vol. 1, no. 1, pp. 58–68, Feb. 2023.
- [17] S. Oslund, C. Washington, A. So, T. Chen, and H. Ji, "Multiview robust adversarial stickers for arbitrary objects in the physical world," *J. Comput. Cognit. Eng.*, vol. 1, no. 4, pp. 152–158, Sep. 2022.
- [18] A. Paramarthalingam and M. Thankanadar, "Extraction of compact boundary normalisation based geometric descriptors for affine invariant shape retrieval," *IET Image Process.*, vol. 15, no. 5, pp. 1093–1104, Apr. 2021.
- [19] S. Han, H. Ma, P. Zhang, and T. Fingscheidt, "Improved MPEG-4 high-efficiency AAC with variable-length soft-decision decoding of the quantized spectral coefficients," *China Commun.*, vol. 16, no. 10, pp. 65–82, Oct. 2019.
- [20] Z. Liu, Q. Li, X. Chen, C. Wu, S. Ishihara, J. Li, and Y. Ji, "Point cloud video streaming: Challenges and solutions," *IEEE Netw.*, vol. 35, no. 5, pp. 202–209, Sep. 2021.



- [21] P. Kudumakis, T. Wilmering, M. Sandler, V. Rodriguez-Doncel, L. Boch, and J. Delgado, "The challenge: From MPEG intellectual property rights ontologies to smart contracts and blockchains [standards in a nutshell]," *IEEE Signal Process. Mag.*, vol. 37, no. 2, pp. 89–95, Mar. 2020.
- [22] O. Derrien, "Detection of genuine lossless audio files: Application to the MPEG-AAC codec," *J. Audio Eng. Soc.*, vol. 67, no. 3, pp. 116–123, Feb. 2019.
- [23] F. Luo, T. Zhou, J. Liu, T. Guo, X. Gong, and J. Ren, "Multiscale diff-changed feature fusion network for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5502713.
- [24] S. UmaMaheswari and V. SrinivasaRaghavan, "Lossless medical image compression algorithm using tetrolet transformation," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 3, pp. 4127–4135, Mar. 2021.
- [25] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, and H. F. A. Hamed, "Hiding data using efficient combination of RSA cryptography, and compression steganography techniques," *IEEE Access*, vol. 9, pp. 31805–31815, 2021.



**XI WANG** was born in Hubei, China, in March 1986. He received the master's degree in art from the China University of Geosciences, in 2013. He is currently a part-time Tutor of landscape design with the School of Art, Hubei University. He is also with Wuhan Design Consulting Group Company Ltd. His research interests include architectural design and landscape design. He received more than seven local, provincial, and national awards for his work in architecture, urban design, and landscape architecture. At present, it has obtained four utility model patents, and four software copyrights.



**KUN YU** was born in Zhongxiang, Hubei, in August 1977. He received the bachelor's degree in fine arts education from the Hubei Academy of Fine Arts, in 2001, and the master's degree in civil engineering from the China University of Geosciences, in 2019.

He was with the Department Manager, Hubei Dolphin Cartoon Company Ltd., from 2001 to 2004. Since 2004, he has been the Director of the Department of Digital Media Arts, Wuchang University of Technology. He has published two papers, wrote one textbook, presided over one provincial scientific research project, and participated in five projects. His research interests include 3D animation and digital media art.



**YIMING JI** was born in Xiaogan, Hubei, in April 1986. He received the bachelor's and master's degrees in animation from the Hubei Academy of Fine Arts, in 2008 and 2012, respectively.

Since 2012, he has been a full-time Teacher with the Zhongnan University of Economics and Law. His academic situation has one SCI paper and one horizontal project. His research interest includes animation. His work won the Excellence Award at the 2016-2017 Busan International Environmental Art Exhibition.

• • •