CLRMA: Compact Low Rank Matrix Approximation for Data Compression

Junhui Hou, Lap-Pui Chau, Nadia Magnenat-Thalmann, and Ying He

Abstract—Low rank matrix approximation (LRMA) is a powerful technique for signal processing and pattern analysis. However, the performance of existing LRMA-based compression methods are still limited. In this paper, we propose compact low rank matrix approximation (CLRMA), a very effective tool for data compression, which extends the LRMA by exploring both the intra- and inter-coherence of data samples simultaneously. Technically, under the assistance of prescribed orthogonal transforms (such as discrete cosine/wavelet transform and graph transform), the CLRMA decomposes a matrix into a product of two smaller matrices so that one consists of extremely sparse and orthogonal column vectors, and the other is a transformed coefficient matrix for reconstruction. Numerically, we formulate the CLRMA problem by minimizing the $\ell_0$-norm and orthogonality regularized approximation (or reconstruction) error and solve it by the inexact augmented Lagrangian multiplier method. We demonstrate the efficacy of CLRMA on various types of real-world data, including 3D meshes, image datasets, videos as well as human motion capture data, that is, the proposed CLRMA, in a much more compact form, can produce comparable approximation (or reconstruction) error as LRMA. Moreover, we present a CLRMA-based compression scheme for 3D dynamic meshes, and experimental results show that it outperforms the state-of-the-art scheme to a large extend in terms of compression performance.

Index Terms—data compression, optimization, low rank matrix, 3D dynamic meshes

I. INTRODUCTION

Given a matrix $X \in \mathbb{R}^{m \times n}$ of $n$ columns corresponding to $n$ data samples in $\mathbb{R}^m$, low rank matrix approximation (LRMA), also known as principal component analysis (PCA), subspace factorization, and so on, seeks a matrix $\hat{X} \in \mathbb{R}^{m \times n}$ of rank $k \ll \min(m, n)$ that best approximates $X$ in the least-squares sense. Alternatively, the rank constraint can be implicitly expressed in factored form, i.e., $X \approx \hat{X} = BC$ where $B \in \mathbb{R}^{m \times k}$ and $C \in \mathbb{R}^{k \times n}$.

See Figure 1. Note that this decomposition is not unique, considering $(BA)(A^{-1}C) = BC$ where $A \in \mathbb{R}^{k \times k}$ is any invertible matrix. Thus, the decomposed $B$ is usually constrained to be column-orthogonal, which approximately shrinks the column space of $X$. Such decomposition not only reduces the space complexity and computational cost in a wide variety of applications, but also reveals the inherent structure of the input data. Therefore, LRMA and improved LRMA have received increasing attention in signal processing, pattern analysis, computer vision, etc.

A. Motivation

More specifically, LRMA is widely used for lossy data compression, such as natural images/videos [1], [2], [3], hyperspectral images [14], 3D motion data [15], [16], [17], [18], traffic data [19], [20], [21], etc., in which some type of data samples are represented as column vectors to construct $X$. After the decomposition, only $k(m+n)$ elements are need to be stored instead of $mn$ ones, with controllable distortion induced. However, the disadvantage of such decomposition is: data samples usually exhibit not only intra-coherence (i.e., coherence within each data sample) but also inter-coherence (i.e., coherence among different data samples). LRMA can exploit the inter-coherence well, i.e., using $B$ with much smaller orthogonal columns to represent $X$, but it fails to explore the intra-coherence. Therefore, the intra-coherence still exists in the columns of $B$, which compromises its compression performance [13], [16], [19], [20], [6], [21]. As an example, Figure 2(a) visualizes several column vectors of $B$ when the LRMA is applied to one video sequence, where we can see that the images are locally smooth (i.e., the bright parts), indicating the coherence.

Some methods have been developed to address this issue. For example, in LRAM-based video/image compression [14], [13], [10], [11], video/image encoders were further employed to encode the columns of $B$, in which each column was reshaped back into one image, and then video/image encoders, e.g., H.264/AVC [22], and JPEG2000 [23], were applied.
But this kind of methods requires the data samples to be 
video/image-like (or matrix) format. Vector quantization 
was also used to encode the columns [1], [12], [2], but the training 
progress requires a lot of extra data and is time-consuming. 
Moreover, the trained codebooks may be biased due to the non-
universal training data, so that large distortion may be induced 
for some types of data. Vásá et al. [17], [18] proposed several 
predictors to explore such intra-coherence when LRMA is 
used for compressing 3D dynamic meshes.

### B. Overview of the Proposed CLRMA

Unlike the above-mentioned methods adopting stepwise manner, in this paper, we propose compact low rank matrix approximation (CLRMA), in which the intra- and inter-coherence are simultaneously explored in the view of optimization, leading that comparable approximation error as LRMA is induced, but the matrix $B$ is represented in a much more compact form than that of LRMA. As Figure 3 shows, the CLRMA multiplies a prescribed orthogonal matrix $\Phi \in \mathbb{R}^{m \times m}$ (such as discrete cosine/wavelet transform (DCT/DWT), and graph transform (GT)) to the input matrix $X$ and then factors $\Phi X$ into a product of the sparse and column-orthogonal matrix $B$ and the coefficient matrix $C$. Thanks to its compact representation, the proposed CLRMA is a very effective tool for data compression, which is demonstrated in the compression of 3D dynamic meshes. As an example, Figure 2(b) visualizes the column vectors of $B$ obtained by the proposed CLRMA which produce the same approximation error as to those of LRMA shown in Figure 2(a), where we can see that most elements of $B$ by CLRMA are zero.

### C. Organization of This Paper

The rest of this paper is organized as follows: Section II lists notations throughout this paper and briefly reviews the
background of LRMA and GT. Sections III formulates the CLRMA problem and provides its numerical solver, whose effectiveness is experimentally validated on various types of data in Section IV. Section V briefly reviews previous LRMA-based compression for 3D dynamic meshes before presenting our CLRMA-based compression scheme. Finally, Section VI concludes this paper.

II. PRELIMINARIES

A. Notation

Throughout this paper, scalars are denoted by italic lowercase letters, vectors by bold lowercase letters and matrices by uppercase letters, respectively. For instance, we consider a matrix $A \in \mathbb{R}^{m \times n}$. The $i$-th row and $j$-th column of $A$ are represented by $a_i \in \mathbb{R}^1 \times n$ and $a_j \in \mathbb{R}^{m \times 1}$, respectively. Let $a_{ij}$ be the $(i, j)$-th entry of $A$ and its absolute is $|a_{ij}|$. $A^T$ and $A^\dagger$ are the transpose and pseudoinverse of $A$. We denote by $||A||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}$ the Frobenious norm of $A$. The $\ell_0$-norm $||A||_0$ counts the number of non-zero entries in $A$. Let $Tr(A) = \sum_{i,j} a_{ij}$ be the trace of $A$. $I_k$ is the identity matrix of size $k \times k$. The hard thresholding operator $T_{\lambda}(a_{ij})$ is defined as $T_{\lambda}(a_{ij}) = a_{ij}$ if $|a_{ij}| > \lambda$ and 0 otherwise.

B. LRMA

Given $X \in \mathbb{R}^{m \times n}$, the LRMA problem can be mathematically formulated as

$$\min_{B \in \mathbb{R}^{m \times k}} ||X - BC||^2_F \text{ subject to } B^TB = I_k.$$  \hspace{1cm} (1)

Setting the derivative of the objective function of $C$ to zero, we obtain $C = B^TX$. Since $||A||_F^2 = Tr(XX^T)$ and $Tr(A) = Tr(A^T)$, we can expend the objective function as

$$||X - BC||^2_F = Tr \left\{ (X - BC)(X - BC)^T \right\} = Tr \left( XX^T \right) - 2Tr \left( BCX^T \right) + Tr \left( BCC^TB \right).$$  \hspace{1cm} (2)

Substituting $C = B^TX$ into (2) and dropping the constant term, we obtain the equivalent problem of (1), i.e.,

$$\max_{B} \text{Tr} \left( B^TXX^TB \right) \text{ subject to } B^TB = I_k.$$  \hspace{1cm} (3)

It is well known that the problem in (3) has an optimal solution \cite{24}, which consists of $k$ eigenvectors of $XX^T$, corresponding to the $k$ largest eigenvalues (in magnitude). We refer the readers to \cite{25} for more technical details about LRMA.

C. Graph Transform

To make the paper self-contained, we also briefly introduce the graph transform used in following sections. Let $G$ be an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with $n$ nodes, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. The graph’s adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is given by

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise}. \end{cases}$$  \hspace{1cm} (4)

Fig. 4. (a) 2D DCT and (b) 2D DWT realized by the 4-level “Haar” matrix were applied to columns of $B$ obtained by LRMA, and then 80%, 60%, 40%, and 20% of all transformed coefficients with smallest magnitude were set to zero before performing inverse transforms. The results are successively shown by the lines with “star”, “square”, “diamond”, “pentagram”. The green line with “circle” refers to result by LRMA. We can see that larger reconstruction error is induced when compared to LRMA, and the error increases significantly with the percentage of zero elements increasing. In contrast, the proposed CLRMA on the same data samples shown in Figure (a) produces comparable RMSE to LRMA even when 80% elements of $B$ are zero. Note that the test data samples are from the video “Carphone”, which are described in Section IV-B. Similar results can be observed for other types of data samples.

Its degree matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$, a diagonal matrix, is defined as

$$d_{ij} = \begin{cases} \sum_j a_{ij} & \text{if } i = j \\ 0 & \text{otherwise}. \end{cases}$$  \hspace{1cm} (5)

Then the graph Laplacian matrix $L \in \mathbb{R}^{n \times n}$ is computed as

$$L = \mathbf{D} - \mathbf{A}.$$  \hspace{1cm} (6)

Since $L$ is positive-semidefinite, its eigenvalues $\lambda_1, \cdots, \lambda_j$, are distinct and non-negative real numbers and its eigenvectors are orthogonal real vectors. Therefore, we can express $L$ as

$$L = U_g A U_g^T.$$  \hspace{1cm} (7)

where the orthogonal matrix $U_g \in \mathbb{R}^{n \times n}$ consists of $L$‘s eigenvectors and $A = \text{diag}(\lambda_1, \cdots, \lambda_j)$. This is known as spectral graph theory \cite{26}. Following \cite{27}, we call the matrix $U_g$ graph transform (GT). Similar to the discrete Fourier bases, $U_g$ also possess harmonic behavior, which can decorrelate the signal defined on the graph.

III. PROPOSED CLRMA

A. Problem Statement

Recall that LRMA can effectively exploit the inter-coherence of data samples, but it fails to exploit their intra-coherence. Thus, the intra-coherence is delivered into the columns of $B$. The issue to be addressed is how to effectively exploit such intra-coherence.

It is well-known that some prescribed orthogonal transforms, e.g., DCT, DWT, and GT denoted as $\Phi \in \mathbb{R}^{m \times m}$ ($\Phi^T \Phi = \Phi \Phi^T = I_m$), can decorrelate signals \cite{26, 28}, i.e., producing approximate sparse transformed coefficients. In order to explore the intra-coherence inheriting from the data samples, one may consider to apply $\Phi$ to columns of $B$ following LRMA, but such stepwise manner induces large reconstruction error shown in Figure 4. The reasons are: (i) the prescribed transforms are general, so they cannot well explore the intra-coherence. (ii) columns of $B$ are separately
decorrelated, which cannot preserve the relationship among them (e.g., orthogonality). Therefore, we propose to explore the intra- and inter- coherence simultaneously during the low rank representation, while keeping the approximation error as small as possible, and the proposed algorithm CLRMA is formulated as an optimization problem:

$$
\begin{align*}
\min_{\Phi, C} & \|X - \Phi C\|^2_F, \\
\text{subject to} & \|\Phi^T B\|_0 = \|\Phi^T B\|_0 \leq s.
\end{align*}
$$

(8)

Here, we constrain $\Phi$ to be sparse with respect to $\Phi$, making the derived $B$ adapted to $\Phi$ during optimization. The difference between $\Phi$ and $B$ by LRMA can be observed by comparing Figure 2(a) and Figure 2(c). Furthermore, taking $\|\Phi^T A\|^2_F = \|A\|^2_F$ and with a slight abuse of notation (i.e., replacing $\Phi^T B$ by $B$), we can equivalently rewrite the problem in (8) as

$$
\begin{align*}
\min_{\Phi, C} & \|\Phi^T X - BC\|^2_F, \\
\text{subject to} & \|B\|_0 \leq s.
\end{align*}
$$

(9)

In all, the proposed CLRMA can be interpreted as: the transformed coefficients of data samples are low-rank approximated under the sparsity constraint, in which $\Phi$ and the sparsity constraint can explore the intra-coherence, and the low rank representation can explore the inter-coherence so that the exploration of such two types of coherences of data samples is simultaneously taken into account and balanced through optimization. Regarding $\Phi$, it can be set according to data samples. For example, we can adopt DCT or DWT as the transform matrix for natural images and videos. For data defined on a general graph, we can take the GT matrix.

### B. Numerical Solver

Instead of solving (9) directly, we solve its Lagrangian version:

$$
\begin{align*}
\min_{B, C} & \|Z - BC\|^2_F + \gamma \|B\|_0 \\
\text{subject to} & B^T B = I_k.
\end{align*}
$$

(10)

where $Z = \Phi^T X$, and the regularization parameter $\gamma$ controls the sparsity of $B$: the larger the value of $\gamma$, the sparser the matrix $B$ is.

We adopt the inexact augmented Lagrangian multiplier method (IALM) \cite{29} to solve (10). Introducing two auxiliary matrices $P \in \mathbb{R}^{m \times k}$ and $Q \in \mathbb{R}^{m \times k}$, we rewrite (10) equivalently as

$$
\begin{align*}
\min_{B, C, P, Q} & \|Z - BC\|^2_F + \gamma \|P\|_0 \\
\text{subject to} & P = B, \quad Q = B, \quad Q^T Q = I_k.
\end{align*}
$$

(11)

The augmented Lagrangian form of (11) is given by

$$
\begin{align*}
\arg\min_{B, C, P, Q} & \|Z - BC\|^2_F + \gamma \|P\|_0 + \text{Tr} \left( Y^T_P (B - P) \right) \\
& + \text{Tr} \left( Y^T_Q (B - Q) \right) + \frac{\rho}{2} \left( \|B - P\|^2_F + \|B - Q\|^2_F \right),
\end{align*}
$$

(12)

where $Y_P$ and $Y_Q \in \mathbb{R}^{m \times k}$ are the Lagrange multipliers, and the regularization parameter $\rho$ is a positive scalar.

With initialized $B$, $P$ and $Q$, the IALM solves the optimization problem (12) in an iterative manner (See Algorithm 1 for the pseudocode). Each iteration alternatively solves the following five subproblems:

1) **The C-Subproblem**: Fixing other variables, the C-subproblem is expressed as

$$
\arg\min_C \|Z - BC\|^2_F.
$$

(13)

Obviously, the optimal solution is

$$
C = B^T Z.
$$

(14)

2) **The B-Subproblem**: The B-subproblem is with a quadratic form:

$$
\begin{align*}
\arg\min_B & \|Z - BC\|^2_F + \frac{\rho}{2} \left( \|B - P + Y_P / \rho\|^2_F \\
& + \|B - Q + Y_Q / \rho\|^2_F \right).
\end{align*}
$$

(15)

Equation (15) reaches the minimal when the first-order derivative to $B$ vanishes:

$$
B = \left( 2ZC^T + \rho(P + Q) - Y_P - Y_Q \right) \left( 2CC^T + 2\rho I_k \right)^{-1}.
$$

(16)

3) **The P-Subproblem**: The P-subproblem is written as

$$
\arg\min_P \|P\|_0 + \frac{\rho}{2} \|P - (B + Y_P / \rho)\|^2_F.
$$

(17)

Let $B = B + Y_P / \rho$, and Equation (17) can rewritten in element-wise as

$$
\arg\min_{\{p_{ij}\}} \sum_{i,j} \gamma \|p_{ij}\|_0 + \frac{\rho}{2} \|p_{ij} - \bar{b}_{ij}\|^2_2.
$$

(18)

Furthermore, Equation (18) is equivalent to

$$
\sum_{i,j} \arg\min_{p_{ij}} \gamma \|p_{ij}\|_0 + \frac{\rho}{2} \|p_{ij} - \bar{b}_{ij}\|^2_2.
$$

(19)

It can be easily checked that $p_{ij}$ has an unique solution given by the hard thresholding operator. Thus,

$$
P = \mathcal{T}_{\sqrt{\frac{\rho}{2\gamma}}} \left( \hat{B} \right).
$$

(20)

4) **The Q-Subproblem**: It is expressed as

$$
\arg\min_{Q} \frac{\rho}{2} \|Q - (B + Y_Q / \rho)\|^2_F.
$$

(21)

According to Theorem 1, Eqn. (21) has a closed-form solution

$$
Q = \hat{B} V D^{-1/2} V^T,
$$

(22)
where \( \hat{B} = B + Y_Q/\rho \), the orthogonal matrix \( V \in \mathbb{R}^{k \times k} \) and the diagonal matrix \( D \in \mathbb{R}^{k \times k} \) satisfy the singular value decomposition (SVD) of \( \hat{B}^T \hat{B} \), i.e., \( \hat{B}^T \hat{B} = V D V^T \).

**Theorem 1 (43):** Given \( X \in \mathbb{R}^{m \times n} \) and \( \text{rank}(X) = n \), the constrained quadratic problem:

\[
X^* = \arg \min_{X \in \mathbb{R}^{m \times n}} \frac{1}{2} \|X - A\|_F^2, \quad \text{subject to } X^T X = I_n,
\]

has the closed-form solution, i.e., \( X^* = A \hat{V} \hat{D}^{-1/2} \hat{V}^T \), where \( \hat{V} \in \mathbb{R}^{n \times n} \) is an orthogonal matrix and \( \hat{D} \in \mathbb{R}^{n \times n} \) is a diagonal matrix satisfying the SVD of \( A^T A \).

5) Updating \( Y_P, Y_Q \) and \( \rho \): Finally, we update the Lagrange multipliers \( Y_P \) and \( Y_Q \), and the parameter \( \rho \) as

\[
Y_P = Y_P + \rho (B - P), \quad (23)
\]

\[
Y_Q = Y_Q + \rho (B - Q), \quad (24)
\]

\[
\rho = \min(\rho_{\alpha}, \rho_{\max}), \quad (25)
\]

where \( \alpha > 1 \) speeds up the rate of convergence.

The IALM is guaranteed to converge for convex optimization problems [29]. However, the objective function in [10] and the orthogonality constraint are both nonconvex. As a result, the optimization problem is nonconvex and there is no theoretical guarantee of convergence of the proposed algorithm. However, extensive evaluation on various types of real-world data shows that our method empirically converge well (see Section IV).

**Algorithm 1** The proposed CLRMA via IALM

**Input:** \( X, \Phi, \gamma, k, \rho, \alpha, \rho_{\max} \)

**Output:** \( B, C \)

1. initialize \( B, P, \) and \( Q \) using the leftmost \( k \) columns of \( I_m \)
2. \( Y_P = Y_Q = 0 \)
3. \( Z = \Phi^T X \)
4. for iter \( \leftarrow 1 : K \) do
5. update \( C \) using (13)
6. update \( B \) using (16)
7. update \( P \) using (20)
8. update \( Q \) using (22)
9. update \( Y_P, Y_Q \) and \( \rho \) using (23), (25), respectively
10. end for

IV. EXPERIMENTAL VERIFICATION OF THE PROPOSED CLRMA

In this section, we verify the effectiveness of the proposed CLRMA on various types of data, including 3D meshes, human motion capture data, image datasets as well as videos. The reconstruction error is measured by the Root Mean Square Error (RMSE) between the original data \( X \) and approximate data \( \hat{X} \), i.e.,

\[
\text{RMSE} = \sqrt{\frac{1}{mn} \|X - \hat{X}\|_F^2}. \quad (26)
\]

The percentage of zero elements of \( B \) is denoted by \( p_B \), which is controlled by \( \gamma \).

A. Evaluation on 3D Meshes and Human Mocap Data

We take four 3D dynamic mesh sequences\(^1\) from [31], including “Wheel” (2501 vertices), “Handstand” (2501 vertices), “Skirt” (2502 vertices), and “Dance” (1502 vertices). We extract 150 frames from each sequence. We also take four human mocap sequences from the CMU Mocap Database\(^2\), i.e., “17_10” (boxing), “56_07” (yawn, stretch, run/jog, jump), “85_12” (jumps, flips, breakdance), and “86_05” (walking, jumping, punching). The mocap data consists of 31 joints, and we uniformly extract 400 frames from each sequence. Some samples are shown in Figures 5(a)(b). The values of \( \rho, \alpha, \) and \( \rho_{\max} \) are set to \( 10^7, 1.003, \) and \( 10^{12} \) (resp. \( 10^{-4}, 1.03, \) and \( 10^8 \)) for 3D meshes (resp. human mocap data), respectively.

The orthogonal matrix \( \Phi \) is realized by GT induced in Section II-C which can be computed by the topology of 3D meshes and human motion capture data, respectively. It is also worth noting that CLRMA (resp. LRMA) is separately and

\(^1\)The dynamic meshes used in this paper refer to mesh sequences that share the same connectivity and differ in the positioning of the vertices from frame to frame.

\(^2\)CMU Mocap Database: [http://mocap.cs.cmu.edu/](http://mocap.cs.cmu.edu/)
Convergence of CLRMA, and human motion capture data (the right column). The first row shows the convergence of CLRMA, $p_B = 0.6$ and $k = 30$ (resp. 12) for 3D meshes (resp. human mocap).

![Graphs showing convergence](image)

Equally performed on the three matrices denoted as $X_x$, $X_y$, and $X_z$, corresponding to three dimensions (i.e., $x$, $y$, and $z$) of the mesh (or mocap data), respectively. When computing RMSE using (26), of the mesh (or mocap data), respectively. When computing RMSE using (26), $\hat{X} = [X_x; X_y; X_z]$ and $\hat{X} = [X_x; \hat{X}_y; \hat{X}_z]$.

As Figure 6 shows, the proposed algorithm converges well. At the same $k$, CLRMA produces comparable RMSEs to LRMA even when the value of $p_B$ increases up to 0.8, leading to compact representation.

### B. Evaluation on Image Datasets and Videos

Two image datasets are employed, i.e., the AR dataset and the Fa of FERET dataset. For each dataset, we randomly extract 150 gray images. In order to reduce computational complexity, the selected images from AR and FERET are resized to $64 \times 64$ and $65 \times 75$, respectively. Two videos with YUV format are tested, i.e., “Carphone” and “Hall”. For each video, the Y components of its first 150 frames are used, and the resolution is reduced to $88 \times 72$ for efficient computation. Some data samples are shown in Figures 5(c)(d)(e)(f), respectively. The values of $p$, $\alpha$, and $p_{max}$ are set to $10^{-4}$, 1.05, and $10^{10}$ for all images and videos, respectively.

The CLRAM is validated under two cases: (I) CLRMA-DCT: $\Phi$ is set as the 2D DCT matrix, i.e., the Kronecker product of two 1D DCT matrices; (II) CLRMA-DWT: $\Phi$ is

http://trace.eas.asu.edu/yuv/
realized by the 2D DWT matrix obtained as the Kronecker product of two 3-level (resp. 4-level) Haar matrices for images (resp. videos). The results shown in Figures 7 and 8 verify the convergence and effectiveness of the proposed CLRMA again. Moreover, we can see that for videos, CLRMA-DWT is better than CLRMA-DCT, that is, at the same $k$ and $p_B$ (e.g., 0.8), the reconstruction error by CLRMA-DWT is smaller than that by CLRMA-DCT since the multi-level DWT has greater potential for sparsely representing images with complex textures than DCT [34]. However, the facial images are relatively smooth, and both DWT and DCT can decorrelate the data well, so that the difference between CLRMA-DWT and CLRMA-DCT is very slight.

V. CLRMA-based Compression Scheme for 3D Dynamic Meshes

So far, the effectiveness of the proposed CLRMA has been extensively verified on various data. In this section, we further propose a CLRMA-based compression scheme for 3D dynamic meshes to show that CLRMA is a very effective tool for data compression.

3D dynamic meshes, encoding geometrical variation of moving objects, are widely used in video games, movie production, virtual reality, etc. Specifically, recent emerging research efforts in 3D telepresence and 3DTV are increasing the demand for such kind of data. To achieve efficient storage and transmission, it is necessary to compress them. Fortunately, 3D dynamic meshes exhibit strong spatial (intra-) and temporal (inter-) correlation, making the compression possible.

A. LRMA-based Compression methods for 3D Dynamic Meshes

Over the past decades, amounts of methods have been developed, such as geometry video-based [35], [36], [11], [10], predictive coding-based [37], [38], transform-based [39], [40], LRMA (or PCA)-based [15], [16], [41], [42], [43], [44], [17], [18], etc. Here, we briefly review the LRMA-based ones, which are most relevant to ours. Alex and Müller [15] first adopted PCA to compress 3D dynamic meshes, representing each frame as the linear combination of few PCA basis vectors. Karni and Gostamn [16] improved this scheme by applying second-order linear predictive coding to PCA coefficients to further remove the temporal coherence. Heu et al. [42] proposed an adaptive bit plane coder to encode the basis vectors. We denote the PCA-based methods in [15], [16], [42] by frame-PCA. The other type of PCA is called trajectory-PCA [41], [45], [46], [17], [18], which represents trajectories of vertices as a linear combination of few PCA basis vectors. Sattler et al. [41] proposed clustered PCA to compress 3D dynamic meshes, in which clustering of trajectories and trajectory-PCA were performed simultaneously to explore the spatial and temporal correlation. Váša et al. studied the trajectory-PCA-based compression for 3D dynamic meshes systematically. For example, they proposed predictive coding to encode the basis vectors [44]. They also introduced three types of predictions to explore the coherence among trajectory-PCA coefficients, i.e., parallelogram-based prediction [45], least squares prediction, and radial basis function-based prediction [17]. Recently, they used the discrete geometric Laplacian of an computed average surface to encode the coefficients [18], and best compression performance is achieved.

B. Proposed Compression Scheme

As shown in Figure 9 the proposed compression scheme decomposes the matrix $X_d (d := \{x, y, z\})$ (each column corresponds to the $d$ dimension of one frame) into a sparse basis matrix $B_d$ and a coefficient matrix $C_d$ using CLRMA.

![Fig. 10. Illustration of the coherence within rows of $C_d$.](image-url)
using the analysis for intra-correlation of relative smoothness is verified. This can be easily explained
DCT is separately performed on rows of the row space. To further reduce this temporal correlation, 1D
nonzero entries of quantized locations are separately coded using arithmetic coding.

Particularly, 3D dynamic meshes consist of successive motions of objects with time varying, so that each row of \( \mathbf{X}_d \) corresponding to the \( d \) dimensional trajectory of one vertex is a relatively smooth curve, indicating strong coherence. After the CLRMA, such coherence still exists in rows of \( \mathbf{C}_d \). For example, Figure 10 plots several rows of \( \mathbf{C}_d \), where their relative smoothness is verified. This can be easily explained using the analysis for intra-correlation of \( \mathbf{B} \) of LRMA along the row space. To further reduce this temporal correlation, 1D DCT is separately performed on rows of \( \mathbf{C}_d \), i.e.,

\[
\hat{\mathbf{C}}_d = \mathbf{U} \mathbf{C}^T_d,
\]

where \( \mathbf{U} \in \mathbb{R}^{n \times n} \) stands for the 1D DCT matrix. Then, the nonzero entries of quantized \( \mathbf{B}_d \) and \( \hat{\mathbf{C}}_d \) as well as their locations are separately coded using arithmetic coding.

Unlike the existing LRMA-based methods mentioned in Section V-A that explore intra- and inter-correlation in a stepwise manner using predictive coding, our scheme explores the two types of correlations simultaneously in the view of computational optimization so that higher compression performance is achieved, which is demonstrated as follows.

**C. Evaluation of Compression Performance**

We evaluate the compression performance of the proposed scheme in this subsection. The compression distortion is measured by the widely-adopted KG error \( [46] \), defined as

\[
\text{KG error} = 100 \times \frac{\| \mathbf{X} - \hat{\mathbf{X}} \|_F}{\| \mathbf{X} - \mathbf{E}(\mathbf{X}) \|_F},
\]

where \( \mathbf{E}(\mathbf{X}) \) is an average matrix of the same size as \( \mathbf{X} \), of which the \( j \)-th column is \((\mathbf{X}_d)_j[1 \cdots 1] \mathbf{X}_y [1 \cdots 1] (\mathbf{X}_z)_j[1 \cdots 1] \mathbf{T} \) with \((\mathbf{X}_d)_j \) is the average of \((\mathbf{X}_d)_j \). The bitrate is measured in bit per frame per vertex (bpfv). The overall compression performance of our scheme is affected by three parameters, i.e., \( k \) (the number of basis vectors of \( \mathbf{B} \)), \( p_B \) (the percentage of zero elements of \( \mathbf{B} \)), and the quantization parameter. Currently, we use exhausted search to determine the optimal combination of them. One can also employ the method in \([46]\) or build rate and distortion models in terms of them like \([10]\) to speed up this process in practice. We compare with Váša’s scheme in \([18]\), PCA-based, which represents the state-of-the-art performance.

Figure 11 shows typical rate-distortion (R-D) curves for our scheme as well as Váša’s, where it can be seen that for “Dance” and “Skirt”, our method produces much smaller distortion than Váša’s at the same bpfv, especially, at relatively small bpfvs. With respect to “Handstand” and “Wheel”, the R-D curves of Váša’s are not given since Váša’s method can only work on manifold-meshes, but sequences “Handstand” and “Wheel” consists of non-manifold-meshes. Our scheme is independent of topology, demonstrating its advantage. Finally, some visual results are shown in Figures 12 and 13 to further demonstrate the performance of our scheme.

**VI. Conclusions and Future Work**

We have presented a new algorithm, namely CLRMA, for data compression. The proposed CLRMA can well explore the intra- and inter-coherence among data simultaneously by low-rank approximating the data with extremely sparse basis functions. The CLRMA is theoretically modeled and solved via iterative optimization. Extensive experiments on various real-world data including 3D meshes, image datasets, human mocap data as well as videos demonstrate its convergence and usefulness, making it a very effective tool for compression. Moreover, a CLRMA-based compression framework for 3D dynamic meshes is developed. Thanks to the computational
optimization of CLRMA, temporal (inter-) and spatial (intra-)
coherence among 3D dynamic meshes are deeply exploited,
leading that at the same bit rate, up to 53% distortion can be
reduced compared to the state-of-the-art method.

In the future, it would be interesting to investigate the
potential of CLRMA for compressing other types of data,
e.g., EEG signals. We also believe the proposed CLRMA has
potential for other applications, e.g., patch-based image/video
denoising, where both the similarity among patches (i.e., low
rank characteristic) and spatial structure of patches (i.e., the
sparseness of patches with respect to particular bases) can be
jointly taken into account.

ACKNOWLEDGMENT

The authors would like to thank Dr. Libor Váša for pro-
viding the rate-distortion data shown in Figures 11 (a) and (c).

REFERENCES

[1] J.-F. Yang and C.-L. Lu, "Combined techniques of singular value decom-
position and vector quantization for image coding," IEEE Transactions on
Image Processing, vol. 4, no. 8, pp. 1141–1146, 1995.
[2] Z. Gu, W. Lin, B.-S. Lee, and C. Lau, “Low-complexity video coding
based on two-dimensional singular value decomposition,” IEEE Trans-
actions on Image Processing, vol. 21, no. 2, pp. 674–687, 2012.
[3] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component
analysis?” Journal of the ACM, vol. 58, no. 3, p. 11, 2011.
[4] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of
subspace structures by low-rank representation,” IEEE Transactions on
Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 171–184,
2013.
[5] D. Meng, Z. Xu, L. Zhang, and J. Zhao, “A cyclic weighted median
method for l1 low-rank matrix factorization with missing entries,” in
Proc. AAAI, 2013, pp. 1–7.
[6] J. Hou, L. Chau, N. Magnenat-Thalmann, and Y. He, “Human motion
capture data tailored transform coding,” IEEE Transactions on Visual-
ization and Computer Graphics, vol. 21, no. 7, pp. 848–859, 2015.
[7] Y. Dai, H. Li, and M. He, “Projective multiview structure and motion
from element-wise factorization,” IEEE Transactions on Pattern Analysis
and Machine Intelligence, vol. 35, no. 9, pp. 2238–2251, 2013.
[8] ——, “A simple prior-free method for non-rigid structure-from-motion
factorization,” International Journal of Computer Vision, vol. 107, no. 2,
pp. 101–122, 2014.
[9] X. Zhou, C. Yang, H. Zhao, and W. Yu, “Low-rank modeling and
its applications in image analysis,” ACM Computing Surveys (CSUR),
vol. 47, no. 2, p. 36, 2014.
[10] J. Hou, L.-P. Chau, N. Magnenat-Thalmann, and Y. He, “Compressing
3-d human motions via keyframe-based geometry videos,” IEEE Trans-
actions on Circuits and Systems for Video Technology, vol. 25, no. 1,
pp. 51–62, Jan 2015.
[11] J. Hou, L.-P. Chau, M. Zhang, N. Magnenat-Thalmann, and Y. He,
“A highly efficient compression framework for time-varying 3-d facial
expressions,” IEEE Transactions on Circuits and Systems for Video
Technology, vol. 24, no. 9, pp. 1541–1553, Sept 2014.
[12] H. Ochoa and K. Rao, “A hybrid dwt-svd image-coding system
(hdwtsvd) for monochromatic images,” in Proc. SPIE Image and Video
Communications and Processing, 2003, pp. 1056–1066.
[13] C. Chen, J. Cai, W. Lin, and G. Shi, “Incremental low-rank and sparse
decomposition for compressing videos captured by fixed cameras,”
Journal of Visual Communication and Image Representation, vol. 26,
pp. 338–348, 2015.
[14] Q. Du and J. E. Fowler, “Hyperspectral image compression using

Fig. 12. Visual results. (a) Original frames uniformly extracted from “Dance”. (b) Left part: decompressed frames with the bpfv equals to 0.25; Right part: rendered distortion map. (c) Left part: decompressed frames with the bpfv equals to 0.55; Right part: rendered distortion map. Note that the distortion map is computed by RMS with respect to the bounding box diagonal.
Fig. 13. Visual results. (a) Original frames uniformly extracted from “Skirt”. (b) Left part: decompressed frames with the bpfv equals to 0.25; Right part: rendered distortion map. (c) Left part: decompressed frames with the bpfv equals to 0.45; Right part: rendered distortion map.
sequences using graph wavelet filter banks,” in Proc. IEEE ICASSP, 2014, pp. 6152–6156.

[40] C.-H. Kwak and I. V. Bajic, “Hybrid compression of dynamic 3d mesh data,” in Proc. IEEE ICASSP, 2014, pp. 6142–6146.

[41] M. Sattler, R. Sarlette, and R. Klein, “Simple and efficient compression of animation sequences,” in Proc. ACM SIGGRAPH/Eurographics SCA, 2005, pp. 209–217.

[42] J.-H. Heu, C.-S. Kim, and S.-U. Lee, “Snr and temporal scalable coding of 3-d mesh sequences using singular value decomposition,” Journal of Visual Communication and Image Representation, vol. 20, no. 7, pp. 439–449, 2009.

[43] L. Váša and V. Skala, “Coddyac: Connectivity driven dynamic mesh compression,” in Proc. IEEE 3DTV, 2007, pp. 1–4.

[44] ———, “Cobra: Compression of the basis for pca represented animations,” Computer Graphics Forum, vol. 28, no. 6, pp. 1529–1540, 2009.

[45] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: measuring error on simplified surfaces,” Computer Graphics Forum, vol. 17, no. 2, pp. 167–174, 1998.

[46] O. Petřík and L. Váša, “Finding optimal parameter configuration for a dynamic triangle mesh compressor,” in Articulated Motion and Deformable Objects. Springer, 2010, pp. 31–42.