Robust Transmission of Motion Capture Data using Interleaved LDPC and Inverse Kinematics

Antonio Carlos Furtado[1], Irene Cheng[1], Frederic Dufaux[2], Anup Basu[1]

[1]University of Alberta, Canada
[2] LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay, Paris, France

Abstract

Recent advances in smart-sensor technology have improved precision in Motion Capture (MoCap) data for realistic animation. However, precision also imposes challenges on bandwidth. While research efforts have focussed on MoCap compression in recent years, little attention has been given to lossy transmission taking advantage of the human perceptual threshold, which allows many online applications, e.g., interactive games, on-demand broadcast, movies and tutoring using dynamic motion sequences. Given the growing applications on mobile devices and wireless networks, associated with insufficient bandwidth, unreliable connection and potential interference or shadowing, data loss is inevitable. We introduce a new Representation for MoCap data, integrating Interleaved Low-Density Parity-Check (I-LDPC), with Keyframe-based Interpolation and Inverse Kinematics, to better address the problem of MoCap data loss during transmission. We believe this is the first study to address robust transmission of MoCap data considering loss. Experimental results assessed using mean opinion scores demonstrate that our approach achieves substantial improvement over alternative transmission methods.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction and Related Work

Motion capture (MoCap) data has been used extensively in the entertainment industry, in particular movies and games. As a result of the latest smart-sensor technologies, MoCap data has achieved higher precision leading to more appealing and realistic animations. A major challenge lies in transmitting the large amount of MoCap data while preserving real-time quality. In addition to traditional on-demand applications, mobile client interfaces using low and unreliable networks have also been developed in recent years. There are applications that require live-streaming of MoCap data, such as content delivery networks (CDNs). Given this diverse range of requirements, we cannot assume that it is always possible to transmit data over reliable connections. Robust transmission of multimedia data has been explored in the past [CYDB08,CB07,AAR05]. However, little attention has been given to lossy transmission of MoCap data. We introduce a MoCap data transmission method that aims to minimize the errors caused by packet loss resulting from unreliable transmission. Our method operates in near real-time, and offers an alternative for streaming data. Our method does not need to retransmit lost packets, avoiding extra overhead on the network. Instead, we minimize perceptual loss using successfully received packets. This is achieved by applying a new Inverse Kinematics MoCap representation coupled with Bezier Interpolation and Foward Error Correction.

New data transmission methods targeting MoCap data compression are still emerging. A big advancement was the creation of a standard interchange format by the MPEG group, called Bone Based Animation (BBA) [ISO01], which can be used to represent any articulated model. However, when it comes to encoding, BBA is restricted to the bit-stream syntax definition. Thus, different encoding techniques have been developed [Pe07,Ce07]; [Pe07] proposed the first known MPEG-4 BBA encoder implementation, while [Ce07] introduced a method that optimizes the power consumption for the device. The limitation of these methods is that they are not designed for unreliable networks, and only focus on improving the bitrate using a TCP connection, which implies higher retransmission rates in case of data loss. Our approach focuses on lossy compression given a time budget and an unreliable network.

The next section describes the proposed robust MoCap transmission strategy, followed by experimental results in Section 3.

2. Proposed Robust MoCap Transmission Strategy

Some applications may prefer lossless transmission, but with the rapid development of mobile applications operating with low bandwidth and in environments with interference or shadowing, lossy transmission is inevitable. Till now the issue of data loss during MoCap transmission has not received sufficient attention, largely

DOI: 10.2312/egsh.20161000

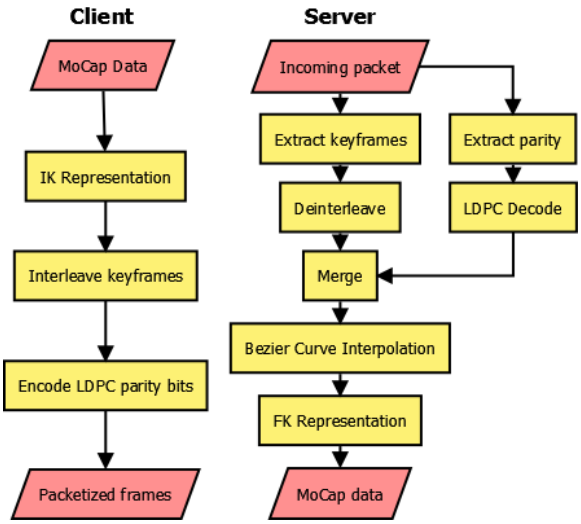

Figure 1: *Integrated transmission strategy.*

due to the limited amount of motion data relative to high bandwidths. The higher precision achieved by the latest smart-sensor technologies has changed this assumption and a large amount of MoCap data can easily overload the available bandwidth. There is an added challenge because the hierarchical structure used by the current Forward Kinematic (FK) representations (commonly adopted by MoCap formats) can lead to significant propagation error at child joints when transmitting over an unreliable channel. To address this problem, we propose an integrated approach. First, we replace the FK with an Inverse Kinematics (IK) representation. Since IK uses positional data, it does not suffer from error propagation. To preserve visual quality, we apply Bezier cubic splines as an efficient keyframe interpolation step. However, packet loss is known to be bursty; loss of important features in adjacent packets can cause perceptually significant distortion. Thus, we leverage our strategy using Forward Error Correction (FEC) by adding protective bits on top of the interleaving protection [CYB12].

The rationale behind our integrated strategy is that although the techniques above address relevant issues, none of them can offer robust MoCap transmission in isolation. Our integrated approach is built upon a Client-Server architecture as illustrated in Figure 1. The client sends interleaved keyframes using positional data, which avoid error propagated down the joint hierarchy. On the other end, the server decodes the packetized data based on the known skeleton model. Missing keyframes are reconstructed using belief propagation (BP) decoding, for which we input Low-Density Parity-Check (LDPC) syndrome bits together with the interpolated data. Finally, filtered trajectories obtained from this process are converted back to a FK representation. In our method, packets are defined using the following structure:

```
struct PACK_FRAGMENT {
    int joint_id;
    long long key_time;
    TRANSFORMATION_TYPE key_type;
    float val_x,val_y,val_z;
};
```

where *joint_id* is a joint identification, *key_time* is a time stamp, $key\_type \in \{ROTATION, TRANSLATION\}$ is a transformation type and the set $\{val\_x, val\_y, val\_z\}$ represents keyframe values, which are transmitted by using an IK representation. It is possible to group keyframes in one fragment using two criteria. First, transformation is associated with up to 3 keyframes. Second, keyframes are synchronized, according to a specific frame rate. Since keyframes are synchronized, a set of packet fragments can also be represented as a list of pairs of joints j and time stamps t:

$$\begin{matrix} (j_1,t_1), & (j_2,t_1), & \ldots, & (j_M,t_1) \\ & \ldots & & \\ (j_1,t_N), & (j_2,t_N), & \ldots, & (j_M,t_N) \end{matrix}$$

where M represents the total number of articulated joints and N is the number of frames in the clip. Since each row in this sequence can be seen as a frame from the clip, and each column contains the set of keyframes for a particular joint, a simple way of interleaving this sequence is by shuffling the rows. Groups of frames are shuffled according to an interleaving window D_I, used to delimit the shuffling range $[t_i, t_{i+D_I}] \quad 1 \leq i \leq N$. This parameter affects both the encoder and the decoder, as it introduces delays on both sides of the transmission. The delay at the encoding side is equal to $D_I/FrameRate$ frames, as the encoder needs D_I frames recorded before being able to shuffle. Similarly, the decoder needs to wait for a corresponding time, as it does not know the arrival order beforehand. The total delay D_T introduced by interleaving is approximately $2(D_I/FrameRate) + TransmissionTime(D_I)$.

2.1. Interleaved LDPC

Our strategy is to apply our interleaving technique and exploit Low-Density Parity-Check (LDPC) codes as a means of transmitting redundant data. While the use of FEC for compressing binary sources has been discussed [Mur01, AG02], to the best of our knowledge we are the first to apply this technique in MoCap compression. In our work, we extend LDPC to send redundant keyframe data. Our process is formulated as follows:

- Let $F^t = (x^t, y^t, z^t)$ be a transformation at time t and $F^t_{(2)} = [x^t_{(2)}|y^t_{(2)}|z^t_{(2)}]$ be its binary representation.
- Let H represent a regularly distributed parity-check matrix, corresponding to a linear code (n,k).

Given our transformation F^t, we can obtain the syndrome bits $Z^t[length(n-k)]$ by multiplying our input transformation by H, such that $F^t_2 \cdot H = [F^t_2|Z^t]$. The syndrome bits Z^t contain the compressed data to be transmitted. To accommodate these bits to a packet fragment, we designed the following fragment extension in our implementation:

```
struct LDPC_FRAGMENT : PACK_FRAGMENT {
    std::bitset<n-k> Z;
};
```

Given that each fragment corresponds to a unique time t, our goal is to add syndrome bits for fragments corresponding to the same

joint, but with different time stamps. This means that transformation F^t is added to the same fragment as the syndrome Z^{t+D_L}, for which D_L is an offset parameter. Similar to D_I, D_L can also cause a delay. To ensure that our previously estimated delay remains unaltered, we add the constraint $D_L \leq D_I$.

2.1.1. Decoding and Reconstruction

The decoding process is started by extracting keyframe information from fragments. Simultaneously, syndrome bits Z are also obtained from the same fragments. Once the loss of a fragment for a given joint at time t is detected, we reconstruct missing keyframes by using an estimate $\bar{F}^t$. This estimate is obtained by evaluating corresponding animation curves (i.e., curves for x, y and z coordinates) at time t. After obtaining a good approximation $\bar{F}^t$, we proceed by feeding this information into a LDPC belief propagation decoder, along with Z^t, as follows:

$$F_2'^t = BPDecode(L([\bar{F}^t_{(2)}|Z^t]))$$

where L represents the log-likelihood ratio (LLR) for each bit in $[\bar{F}^t_{(2)}|Z_t]$. L is a bitwise operation, defined as:

$$\ln\left(\frac{Pr(b_i = 0|y)}{Pr(b_i = 1|y)}\right)$$

where $Pr(b_i = 0|y)$ is the conditional probability of bit b_i being 0, given the received vector y. The LLR calculation is defined for each bit as follows:

- For bits representing Z^t: We assume these bits to be sent through an error-free channel. Therefore, we are certain that they contain the correct value. In this case, $LLR(0) = +\infty$, whereas $LLR(1) = -\infty$;
- For bits representing $\bar{F}^t_{(2)}$: As $\bar{F}^t$ is actually represented as x^t, y^t, z^t, we can represent it as a set of 3 float values. For each float, it follows that $|LLR(b_i)| < |LLR(b_{i+1})|$, with i representing the significance order of b. This constraint is applied because it is less likely that more significant bits will differ between the interpolated transformation $\bar{F}^t$ and the original F^t.

After decoding the LLR sequence, we obtain a reconstructed estimate F'^t. By decoding its value, we expect this estimate to be closer to the original value than the interpolated estimate $\bar{F}^t$. F'^t is then added to the curve as an extra keyframe. Finally, all the keyframes, including the reconstructed ones, are converted to a FK representation.

3. Experimental Results

Since we are not aware of any prior work that addresses unreliable transmission of motion capture data, we evaluate the performance of our approach by comparing three different strategies for transmission:

- **Simplified Serialized Transmission:** In this approach, a traditional hierarchical skeleton has its keyframes packetized and transmitted in the same order as they are displayed. Neither interleaving nor LDPC is applied;
- **Interleaved Transmission:** Interleaving is added to the simplified transmission;
- **Interleaved LDPC (I-LDPC) Transmission:** Here we apply our proposed method, illustrated in Figure 1. Interleaving window D_I and LDPC offset D_L are specified for each test case.

Method / Clip	03_03	60_01	85_14
Simple	3.6	3.5	2.3
Interleaved	3.6	3	3
I-LDPC	**4**	**4**	**4.1**

Table 1: *The average mean opinion scores show that our method (I-LDPC) delivers better perceptual qualities than the other two methods. Ratings used is from* 1 *(worst) to* 5 *(best). Clip* 85_14 *contains most dynamic motion and benefits more from our approach.*

We performed tests on three motion clips extracted from the CMU Graphics Library, namely 03_03, 60_01 and 85_14. Clip 03_03 presents a slow transition between poses, and there is not much movement for most of the joints. Whereas clip 85_14 is the exact opposite. For this clip the transitions are fast and joints are highly articulated. Clip 60_01 falls between these two. By selecting these clips, our goal is to demonstrate how loss affects motion at different transition speeds. The same conditions and execution parameters were applied to all the test cases at a frame rate of $30fps$, and the interleaving window was set to 90 and 15 frames respectively. The number of syndrome bits Z^t was set to be equal to the number of bits used to represent a transformation $F^t_{(2)}$. Results for Clip 85_14, with 74% loss and average burst length of 35 packets are shown in Figure 2, which illustrates that our I-LDPC algorithm produces smooth transitions that are similar to the original motion. Additional examples are shown in the supplementary video.

We conducted evaluation tests with human subjects to determine how "close" the decoded MoCap data is to the original data. Since Root Mean Squared Error (RMSE) is inadequate to measure the perceptual closeness of the reconstructed motion [Ari06], we follow the mean opinion score approach [FCB11], with 12 participants. The results are reported in Table 1. We can observe that our I-LDPC method has a higher rating in all the tests, especially for the most dynamic motion. This demonstrates that animations with faster transitions between poses are highly sensitive to the adverse effect of data loss, and can benefit more from our optimized transmission.

4. Conclusion

We presented a new IK Representation for MoCap data to optimize transmission over unreliable channels. Given packet loss, our proposed Interleaved LDCP (I-LDPC) method preserves quality better based on an integrated strategy, extending LDCP, coupled with Bezier interpolation and FEC to provide optimal bit protection. Our approach does not require retransmission avoiding extra network overhead. User studies showed the perceptual gain in the quality of the motion using I-LDPC over alternative approaches, especially for animations that contain faster transitions between poses. In future work, we intend to assess the benefit of applying variable weights on joints to compensate the corresponding deviation error generated from Bezier interpolation, and perform a larger scale user evaluation. We will also apply our techniques to create

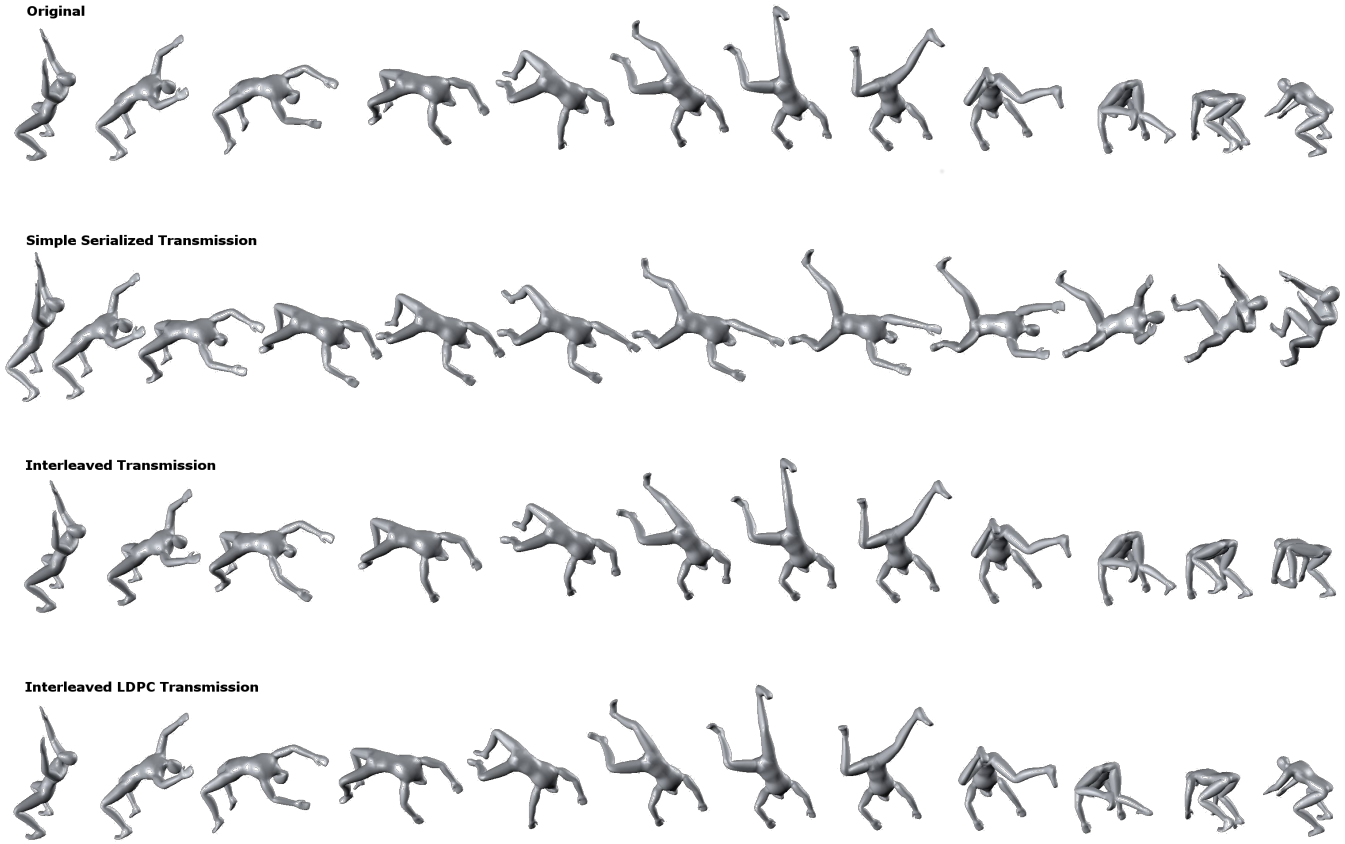

Figure 2: *For the simplified transmission, rotation information from the root joint is severely affected during the backflip, causing major distortion. Quality is improved when interleaving is introduced, on the third row. However, the last pose still contains some distortion for the leg. The I-LDPC method, shown on the last row, is the one that is closest to the original.*

dynamic model-based coding characters [YB99, YB01] at very low bit rates.

References

[AAR05] ALREGIB G., ALTUNBASAK Y., ROSSIGNAC J.: Error-resilient transmission of 3d models. *ACM Transactions on Graphics* (2005), 182–208. 1

[AG02] AARON A., GIROD B.: Compression with side information using turbo codes. In *Data Compression Conference, 2002. Proceedings. DCC 2002* (2002), IEEE, pp. 252–261. 2

[Ari06] ARIKAN O.: Compression of motion capture databases. *ACM Transactions on Graphics* (2006), 890–897. 3

[CB07] CHENG I., BASU A.: Perceptually optimized 3D transmission over wireless networks. *IEEE TMM* (2007), 386–396. 1

[Ce07] CHATTOPADHYAY S., ET AL.: Model-based power aware compression algorithms for mpeg-4 virtual human animation in mobile environments. *IEEE TMM* (2007), 1–8. 1

[CYB12] CHENG I., YING L., BASU A.: Perceptually coded transmission of arbitrary 3d objects over burst packet loss channels enhanced with a generic jnd formulation. *Selected Areas in Communications, IEEE Journal on 30*, 7 (2012), 1184–1192. 2

[CYDB08] CHENG I., YING L., DANIILIDIS K., BASU A.: Robust and scalable transmission of arbitrary 3d models over wireless networks. *Image and Video Processing Journal* (2008). 1

[FCB11] FIROUZMANESH A., CHENG I., BASU A.: Perceptually guided fast compression of 3-d motion capture data. *Multimedia, IEEE Transactions on 13*, 4 (2011), 829–834. 3

[ISO01] ISO: *Information technology—Coding of audio-visual objects—Part 2: Visual*. Tech. rep., 2001. 1

[Mur01] MURAYAMA T.: Statistical mechanics of linear compression codes in network communication. *arXiv preprint cond-mat/0106209* (2001). 2

[Pe07] PREDA M., ET AL.: Optimized mpeg-4 animation encoder for motion capture data. In *International conference on 3D web technology* (2007), pp. 181–190. 1

[YB99] YIN L., BASU A.: Integrating active face tracking with model-based coding. *Pattern Recognition Letters* (1999), 651–657. 4

[YB01] YIN L., BASU A.: Nose shape estimation and tracking for model-based coding. *ICASSP* (2001). 4