

Development of a Human Movement Codec

David Chiasson

Abstract—Real-time kinematic data is central to many emerging technology applications, yet to date few techniques have been proposed to process and represent with this relatively new information modality in an efficient manner. This paper explores techniques for the compression of human kinematic data. Techniques from traditional multimedia networks are compared with several novel approaches. Simple, computationally cheap techniques are shown to losslessly compress kinematic data by over a factor of ten compared with traditional representations. Furthermore, these simple techniques are shown to be superior to more complex approaches. All techniques discussed in this study are implemented and released as open source code which can be quickly integrated into a variety of computational platforms.

Index Terms—kinematic data, human movement, compression, codec

I. INTRODUCTION

MANY emerging technology applications rely on real-time kinematic data as a crucial component. These include virtual reality, autonomous driving, internet of things, physical therapy, wearable electronics, and human performance among others. These applications have been enabled by the recent explosion of cheap inertial based sensors (IMUs) along with mobile computation power to process this data at high sampling frequency. Kinematic data represents a nascent field of multimedia data which is starkly under-developed compared to the existing maturity of text, audio, and visual type data processing techniques. In order to enable the modern applications listed above, efficient and standard techniques for representing and processing kinematic data are needed.

This study explores a range of both traditional and novel compression techniques applied to human kinematic data gathered by 6-axis IMUs. The best performing techniques are identified and discussed. Only lossless compression techniques are considered. The reason for this is that lossy compression inherently involves a value judgment about what information is useful and what information is irrelevant. In the absence of a specific application, no justification can be made for discarding any portion of information.

This study focuses on highly practical approaches which can be used in modern applications. To demonstrate this, all compression techniques have been implemented and released as open-source C code using fixed point computations. The hope of the authors is that this code can be a starting point for academic or industry developers implementing some kinematic data application on any type of computational platform.

All compression algorithms can be divided into two phases, modeling and coding[TO CITE] as shown in figure 1. The model incorporates our knowledge about the signal to be compressed. It computes a probability mass function (PMF) which is an estimated likelihood of all possible input symbols. A dynamic model is one which changes its PMF estimate

after new input symbols are received. The coder then reads the uncompressed data and uses the PMF to choose a unique code for each input symbol. Short codes are chosen for likely symbols and long codes are chosen for unlikely symbols such that the total length of the compressed data is minimized. The decompressor uses an identical model to provide the same PMF to the decoder which is used to revert each code back into the original symbol. If the model is dynamic, this output is used to update the model for future predictions. Coding is a solved problem and will thus not be discussed in this work.[TO CITE] Modeling on the other hand has been proven unsolvable and must be revisited for each new class of signal.(due to the pidgeon hole principal?)[TO CITE]. To the author's knowledge, no previous work has directly addressed the modeling problem for kinematic data signals. This provides the motivation for the current work. Modeling has been shown to be equivalent to the artificial intelligence problem [TO CITE].

Since the theoretical limit on the compression provided by the coder is set by Shannon's entropy law

$$H = - \sum_i P_i \log P_i$$

[to cite]

The goal of the model is to produce an accurate PMF with minimum entropy. Intuitively this means that compression is higher the more "sure" the model is about some occurrences over other ones. If an optimal encoder is used, the compression ratio can be regarded as a quantitative measurement of the model's understanding of the underlying signal. Because of this, we expect the performance of various signal models to give us insight into the content of the kinematic signal.

TODO: why do we assume our signal is geometrically distributed about the mean!? TODO: relationship for natural signals of residual signal and compression ratio

In this work, the maximum likelihood occurrence according to the model will be referred to as the prediction. The difference between the prediction and the actual sample will be referred to as the residual. For practical implementation, it is advantageous to convert the original signal into the residual signal before coding so that the coder can be computationally optimized to compute encodings for zero mean random variables (or same distribution? not recalculating).

Define stream? node? anything else?

We hypothesize that compression ratios will be comparable to those achieved in lossless audio compression. We also hypothesize that the best performing codec will utilize some cross axis correlation and be informed by physics based models.

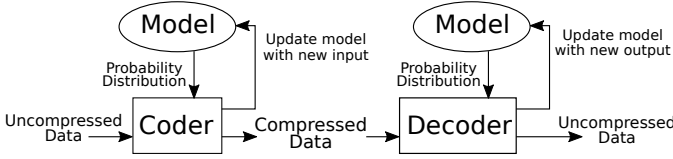


Fig. 1. General compressor and decompressor

II. METHOD

In order to meaningfully and repeatably demonstrate the performance of a compression algorithm, a public kinetic signal corpus must be selected. For this study, the Human Gait Database (HuGaDB) is used. HuGaDB is a public dataset of six-axis IMU signals collected from six different body segments of 18 healthy subjects performing 12 different movement activities[TO Cite]. This database was selected because it allowed the comparison of compression techniques across body segment, subject, and activity in addition to sensor modality.

The data collected from three random subjects was designated as training data, and only data from the remaining 15 subjects was used in our results. The training data was used to train the lasso regression of our cross-stream FIR filter discussed later.

A. Reference Encodings

As it is proven impossible to claim maximum compression [TO CITE] it is necessary to select a reference when evaluating the performance of a compression algorithm. In this study, the baseline encoding (exhibiting a compression ratio of one) is comma separated value (CSV), a simple text based format which is the de-facto standard for storing sensor information. Performance is compared by computing the compression ratio CR relative to CSV via the following formula:

$$CR = \frac{\text{size of CSV file}}{\text{size of compressed file}}$$

To give further context to performance, established algorithms from both text and audio compression are applied to the kinematic data. In total, the following five reference encodings are used to give context to compression ratios:

- **CSV** Standard text based format. To minimize variance in csv size, each data point is formatted to the the same text length resulting in a file size of sixteen bytes per sample.
- **Binary** The optimal fixed size format. In our corpus, every sample is two bytes. This would be the optimal compression if each sample were an IID random variable uniformly distributed across all possible sample readings.
- **Binary Rice Encoded** Binary files with Rice-Golomb encoding[TO CITE] applied. This would be the optimal compression if each sample were an IID geometrically distributed random variable with mean zero.
- **ZIP compression of CSV** ZIP is a general purpose compressed file format integrated into all major computer systems. [TO CITE] ZIP was executed using the DEFLATE to cite] method and a compression level of 6.

- **FLAC compression of binary** FLAC is a popular, open-source lossless audio codec (include options)[TO CITE]. FLAC was executed using the linear predictive coding [TO CITE] technique and a compression level of 5.

B. Tested Encodings

TODO: can I prove that the optimal IIR linear filter cannot exceed that of the optimal FIR linear filter?

Several restrictions are placed on the algorithms considered in this study. First, a viable algorithm must be causal. This is a basic requirement for an algorithm to be real-time. We also chose to only consider algorithms with zero filter delay. Since our sensors operate at a relatively low sampling frequency of 60Hz, a delay of one sample would be 16ms which is significant for modern information networks.

We also only consider algorithms which are node independent. The model for each signal considers at most the information from the six co-located signals including it's own past input. While it is likely that utilizing inter-node correlation could produce better compression ratios, especially if a human biomechanics model is introduced, such an approach would limit the usefulness of said algorithm to that specific placement of nodes on the body.

Finally, only lossless algorithms are considered. As discussed in the introduction, a lossy algorithm requires an objective metric for determining how much data loss and of what type is acceptable. Without this metric it is trivial to create a compression algorithm with an infinite compression ratio but which loses all information.

In practice, the implementation of a strictly lossless algorithm turns out to be non-trivial. This is because the industry standard for floating point computation IEEE 754 [TO CITE] is not sufficiently stringent to guarantee identical results on various implementations [TO CITE]. Additionally, a compiler or interpreter which processes the source code for an algorithm implementation will often utilize mathematical properties such as commutativity to optimize computation. This may result in different machines performing floating point operations in a different order which could lead to differing results even if each rounding operation were well defined by IEEE 754. In order to avoid both of these scenarios and guarantee identical results across any computer, all algorithms in this study are implemented using only integer operations and fixed-point 16.16 precision.[cite fixed point math library?] There is significant cost associated with this technical decision. Namely every computation has the same distribution of rounding error independent of its magnitude. This can lead to significant numerical issues for complex algorithms [TO CITE]. In the context of compression, complex or numerically unstable algorithms may produce compression ratios which underestimate their understanding of the underlying signal since their implementation is not equivalent to their mathematical derivation.

- **Delta encoding** Current sample is predicted to be equivalent to the previous sample so that the difference between the two is encoded. If a signal varies slowly with time, this signal will be smaller than the original signal.

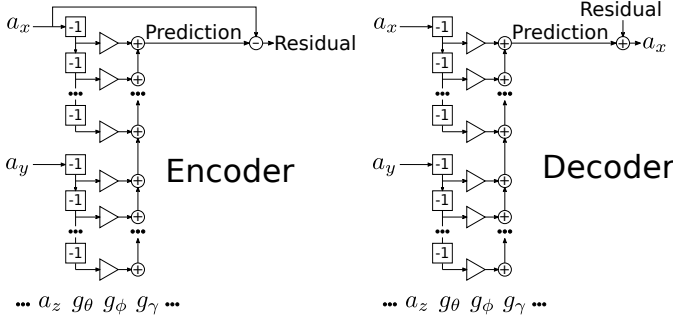


Fig. 2. Flow diagram of encoder and decoder.

- **Linear extrapolation** Current sample is estimated as a linear extrapolation from previous samples. Also known as first order polynomial regression.
- **2nd to 5th order polynomial regression** This technique assumes that the signal is a polynomial which is estimated from a least squares regression of past samples. This polynomial is then extended to get a prediction of the current sample.
- **Spline extrapolation** A spline is the minimum curvature piece-wise polynomial which connects a set of points. It is commonly used for interpolation, namely computer graphics smoothing. This technique was selected as splines are known to avoid Runge's phenomenon which is witnessed when extrapolating higher order polynomials. Results from the cubic spline with natural boundary conditions are presented in this paper.
- **Linearized rotating gravity** A Newtonian physics based model in which a constant magnitude acceleration vector rotates according to gyroscope readings.
- **Trained auto-correlation linear prediction** The optimal linear predictor of training data utilizing each signal's history.
- **Trained cross-correlation linear prediction** The optimal linear predictor of training data utilizing all six co-located signal histories.

C. Implementation

To show algorithm performance in a realistic scenario, as well as to provide tools of benefit to the community, the algorithms in this study have been implemented in the C programming language and released as open-source code found at https://github.com/dchiasson/kinetic_codec. TODO[change the name of the repo, clean up docs, and release]. The choice of programming language as well as the restriction to integer arithmetic allow this code to be easily incorporated into programs executing on either a traditional computer or embedded computation platforms, even those without a floating-point unit.

Figure 2 shows the computational flow chart [probably remove this]

III. RESULTS

Figure 3 shows the best compression ratio of all test data for each encoding type. Delta encoding saw the highest compression of the polynomial regression techniques (CR=14.457),

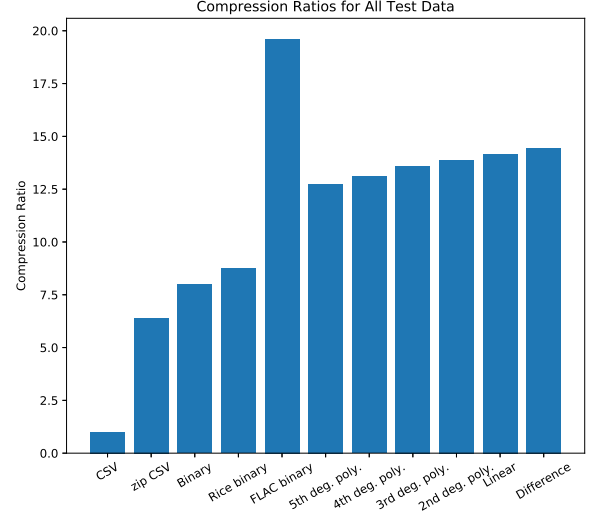


Fig. 3. Compression ratio (original size / compressed size) of each encoding technique on all test data. Larger is better.

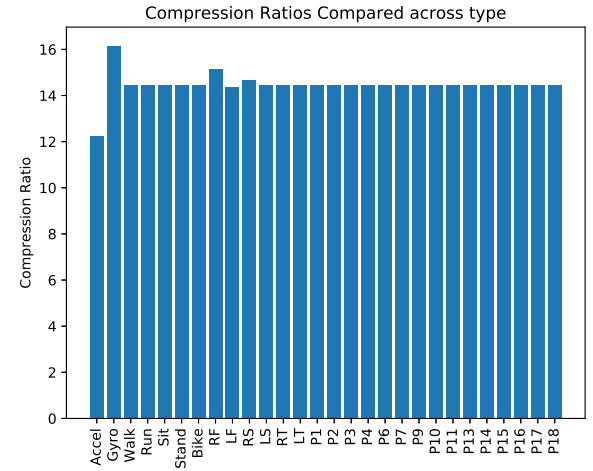


Fig. 4. Compression ratio (original size / compressed size) of each encoding technique on all test data. Larger is better.

and each higher degree polynomial performed progressively worse with 5th degree polynomial at the bottom (CR=12.740).

Table TODO shows the breakdown of CR for each modality, activity, body segment, and subject. No significant difference was found with any algorithm for compression between subject or activity. There are minor differences across body segment and major differences between sensor modality.

TODO: quantify statistical significance

Tables of detailed results k, order: total, accel/gyro, activity, segment, subject

TODO: Discussion of cross stream FIR coefficients, pole zero plots, precision effect

IV. DISCUSSION

These results show that even simple techniques result in a significant compression over the traditional encoding of CSV.

	K	Order	All	Activity					
				Walk	Run	Sit	Stand	Bike	
CSV	-	-	1	1	1	1	1	1	
zip CSV	-	-	6.39430101906924						
Binary	-	-	8	8	8	8	8	8	
Rice binary	12	-	8.75550182215849	8.7555021779827	8.75551024333942	8.75550182215849	8.75550283269934	8.75550315181756	
FLAC binary	0	0	19.6099365007281						
Spline	0	0							
5th deg. poly.	8	19	12.7401178924573	12.7402049974879	12.7409038571714	12.7401953539615	12.7402467059078	12.7404859033232	
4th deg. poly.	7	19	13.1118619820637	13.1120129435405	13.1132114972774	13.1119919557271	13.1120837439905	13.1124890829694	
3rd deg. poly.	7	19	13.5969932983155	13.5971381903649	13.5983046256314	13.5971201261144	13.5972121298592	13.5975961923989	
2nd deg. poly.	7	13	13.878393070373	13.8785077289018	13.8794193164099	13.8784928430834	13.878563302906	13.8788698732695	
Linear	7	8	14.1599487678905	14.1600319270935	14.16070287805	14.1600215034495	14.1600743663739	14.1603025161539	
Difference	7	1	14.4571554136452	14.4571827675638	14.4574186266642	14.457181748902	14.4571975915343	14.4572703883325	

TABLE I

THIS TABLE IS WAY TO BIG AND NEEDS TO BE REFORMATTED

If bandwidth or storage space is any concern in a kinematic data application, using a binary format would have very few disadvantages other than the loss of human-readability. Even more significant gains can be achieved with the trivial technique of delta encoding.

We were surprised by the small variation in CR across body segments, and the complete lack of variation across activity and subject. The small differences of body segments may even be due not to the differences in movement, but due to noise and bias differences between sensor hardware, as each trial had the same sensor on the same body segment (can I confirm this?). The activity and subject results average across sensor hardware, but body segment would expose any differences.

This observation, along with the surprisingly poor performance of our physics based model suggests that the information in our signal has less to do with the human body and it's movement than previously assumed. If the movement of the human body made up the vast majority of the signal content, we would expect to see significant differences in compression ratios between running and sitting. It is possible that the bulk of the entropy in our signal is due to noise, sculling and/or coning. If this is the case, it presents a significant opportunity for lossy compression algorithms to achieve significantly more compression than shown in this study.

An alternate explanation for this result may be the aliasing of the kinematic signal. 60Hz is common for consumer grade IMUs, but it is generally agreed to be well below what is required to fully capture human movement [TO CITE]. Fast movements such as running are often sampled at 1000Hz. If the signals are significantly undersampled, we may be losing much of the high frequency information that would differentiate fast movements from slow movements. [TODO: research effect of aliasing on information content]

Was poor performance of complex algorithms due to precision issues?

Surprising performance of difference encoding

Lack of correlation between accelerometer and gyroscope

Compression rate compared to audio

A. limitations

The compression ratios presented in this paper are intended to demonstrate the relative difference between compression techniques and may not be representative of the absolute

CR experienced in other applications. There are many other factors which can affect the compression ratio which are not explored in this paper. Namely, sensor hardware differences of precision, noise, bias, and sampling rate are expected to have a large impact on the CR achieved. Additionally, CSV encoding which is used as the reference in this study, does not have a well defined size per sample. While in this study the samples were formatted to a fixed text length to preserve precision and cause deterministic size, the real size of a CSV file could vary dramatically with no upper limit depending on the formatting chosen. (should I leave that part out? I basically admitted that all my CR values are arbitrary...)

In the future, these techniques will be applied to various sensing hardware, sampling rates, and body segments. improvements: dynamic K, blocking for error recovery

V. CONCLUSION

Kabam! Splat!

ACKNOWLEDGMENT

The authors would like to thank your mom.

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.