Research on Lightweight Method of MBD Model for Additive Manufacturing

XiaoXu Wang, Hao Qin*, Zhi Liu, Yu Zhang

Guangdong Institute of Intelligent Manufacturing, Guangdong Key Laboratory of Modern Control Technology, Guangzhou 510070, Guangdong, China

h.qin@giim.ac.cn * Corresponding author: e-mail: 516442184@qq.com

Abstract—In order to solve the problems of difficult storage and slow transmission of large data in remote transmission of 3D model for additive manufacturing, a lightweight method on MBD model is proposed. A lossless compression algorithm is applied and improved in this paper to simplified the product data model and minimize data exchange file. At the same time, a detailed geometric information model is retained to realize the rapid transmission of lightweight model, which provides a convenient method for remote transmission and acquisition of 3D model in additive manufacturing process. A case study on the MBD model of a packaging machine has verified is taken as an example to verify the feasibility of the method.

1. Introduction

With the guidance of "Made in China 2025" strategy, "additive manufacturing" has developed rapidly in different fields, together with relative technologies. Model-based Definition (MBD) model technology is one of them. As the only carrier of data, MBD model can fully express all kinds of detailed information in product manufacturing [1], but its three-dimensional model is relatively complex, and the file takes up a lot of memory space, usually more than 100 MB or even several GB [2-3]. In 3D printing, if remote transmission of information is required, it will cause slow transmission because of the large scale of the 3D model file. This problem results in delay of data and information acquisition as well as the online viewing of the 3D model [4-5]. In order to meet the needs of manufacturing, many scholars at home and abroad have done a lot of research on data lightweight techniques for MBD model.

For the research on MBD model data compression processing, the earliest method is by Taubin and Ouma who have proposed a geometric lossless compression coding algorithm based on grid processing [6-7], which is an efficient coding algorithm to decompose complex mesh model into simple topological structure information compression coding, vertex prediction estimation and compression. In reference [8], Li has proposed an improvement method on the shortcomings of EPRT and EASM file formats, combined the advantages of XML and VRML, as well as exposing shortcomings. From his study, it takes 10 seconds to view some simple models, while longer time for some large and complex models. In reference [5], Wang has proposed a multi-level model lightweight method, which can compress, filter and optimize the original model data through reducing the amount of model data and memory consumption, with a result in speeding up browsing and display. This compression algorithm used is based on Huffman compression algorithm, which has shortcomings on low compression rate and speed. Wang et al. proposed a lossless compression algorithm for complex product model data [9]. The algorithm uses non-geometric information filtering, curve and surface simplification, coding
compression and other operations to process the original product model data. Under the condition of ensuring a certain model accuracy, it has a higher compression ratio and meets the requirements of engineering application.

At home and abroad, most of the data lightweight methods for 3D MBD model aim at the lightweight processing of ontology solid model, while few amount of research on its transmission and storage. However, in the era of big data, the research on the problems of remote transmission of model and large storage space in additive manufacturing technology needs to be solved urgently to accelerate complex manufacturing. To fill this gap, this paper has proposed a data lightweight method based lossless compression algorithm for the remote data transmission for additive Manufacturing.

2. Method
The lightweight of 3D model is mainly divided into two parts, one is the solid lightweight of 3D model, the other is the data lightweight of 3D model. In this paper, we choose lossless compression algorithm to process the data of 3D model, which can meet the requirements of fast transmission, high precision, model without loss.

Compared the advantages and disadvantages of lossless compression algorithm, such as Huffman compression algorithm, Xiangnong van Nuel algorithm [10], adaptive Huffman compression algorithm and LZSS algorithm [11-12]. In this paper, we choose the adaptive Huffman compression algorithm, which uses compression and Huffman tree construction to perform compression encoding and decoding at the same time. And the running time of this algorithm is linear with the length of the input source string, while the storage space will not be determined by the length of the input string. [11,13]. However, this algorithm is the repeated encoding of repeated strings, which leads to the problem of low compression rate, when the data lightweight of MBD model. Therefore, this paper improves the adaptive Huffman compression algorithm by adding a string repeat finder to improve the compression rate. The improved adaptive Huffman compression algorithm is shown in Figure 1.

![Fig. 1 Flow chart of improved adaptive Huffman coding process](image-url)
2.1. String repeat finder

As for how to quickly find duplicate strings, the principle is: set the sequence of input MBD model data as: \( X = x_1 x_2 \ldots x_n \), output as code sequence: \( Y = y_1 y_2 \ldots y_t \), the relationship between \( x_i \) and \( y_t \) meets the following requirements:

\[
Y = \begin{cases} \min \{ k | x_{i+k} = x_{i+k} \} & t = 1 \\ k = x_{i+k} \neq x_{i+k} \end{cases}
\]

If \( y_{i-1} = y_i \), think \( x_i \) is in repeat mode, otherwise it is normal mode. Therefore, if \( y_i = y_{i-1} = \ldots = y_{i-a}, \delta \geq 1 \)

and \( y_i = y_{i-1} \ldots y_{i-\delta} \neq y_{i-\delta+1} \), its corresponding string \( x_{i-\delta+1} \ldots x_{i-1} \) forms a repeating pattern of \( \delta+1 \).

In order to increase the matching length of one time, we can set \( y_i = y_{i-1} = \ldots = y_{i-a} \delta \geq 1 \)

and \( y_i = y_{i-1} \ldots y_{i-\delta} \neq y_{i-\delta+1} \), its corresponding string \( x_{i-\delta+1} \ldots x_{i-1} \) forms a repeating pattern of \( \delta+1 \).

Otherwise, if \( y_i = y_{i-1} = \ldots = y_{i-a} \), \( x_i \) and \( x_{i-1} \) have the same output value, a longer string can be obtained. Add a letter to the letter set to represent the repeated pattern. The prefix code is used to encode the repeating pattern, so that the length of variable length encoding is the shortest and the compression effect is improved. On the contrary, the adaptive Huffman optimal tree is built according to the adaptive Huffman compression algorithm, and the coding is carried out according to the principle of Huffman variable length coding.

2.2. Constructing an improved adaptive Huffman optimal tree

A complete binary tree of the optimal prefix encoding of the character set \( C \), represented by \( T \). The frequency of character \( c \) in \( C \) is \( f(c) \). \( x \) and \( y \) are two leaves and brothers in tree \( T \), and \( z \) is their father. Consider \( z \) as a character with frequency \( f(z) = f(x) + f(y) \), the tree \( T' = T - \{x, y\} \), represents an optimal encoding of character set, namely

\[ C' = C - \{x, y\} \cup \{z\} \]  

The average code length \( B(T) \) of \( T \) can be expressed by the average code length \( d \) of \( T' \). For any \( c \in C - \{x, y\} \), there is

\[ d_t(c) = d_t(c) \]

Therefore:

\[ f(c)d_t(c) = f(c)d_t(c) \]

On the other hand,

\[ d_t(x) = d_t(y) = d_t(z) + 1 \]

Therefore:

\[ f(x)d_t(x) + f(y)d_t(y) = (f(x) + f(y))(d_t(z) + 1) \]

\[ = f(x) + f(y) + f(z)d_t(z) \]

It is concluded that:

\[ B(T) = f(x) + f(y) \]

2.3. The improved adaptive Huffman optimal tree is coded with variable length

There are two forms of encoding: one is that the string belongs to the normal mode, which is encoded in the form of "0" and "1", according to the principle of adaptive Huffman compression algorithm of the left and right subtrees. The other is that the string belongs to repetition pattern, which is encoded
according to the principle of string repeat finder. The prefix is represented by letters in repetition mode, which makes the variable length encoding the shortest and achieves the compression effect.

\[
H(m) = \lim_{n \to \infty} \frac{1}{n} G_n
\]  
(8)

Where \( H(m) \) is the entropy of the improved adaptive Huffman code.

\[
G_n = \sum_{i=1}^{n} \sum_{i=2}^{n} P(X_i = i, X_2 = i, \ldots, X_n = i) \log P(X_i = i, X_2 = i, \ldots, X_n = i)
\]  
(9)

\[
P(X_1 = i_1, X_2 = i_2, \ldots, X_n = i_n) = \frac{W_j}{n}
\]  
(10)

For source characters, the \( m \) set is:

\[
A = \{a_1, a_2, \ldots, a_k\}
\]  
(11)

The probability model is as follows:

\[
\{P(a_1), P(a_2), \ldots, P(a_k)\}
\]  
(12)

The average length of the code is expressed as:

\[
\bar{l} = \frac{1}{k} \sum_{i=1}^{k} P(a_i) l_i
\]  
(13)

Because \( \sum_{i=1}^{k} 2^{-l_i} \leq 1 \), therefore \( H(m) - \bar{l} \leq 0 \), indicates that the lower bound of encoding length is \( H(m) \).

The upper bound is: for a given source character, its weight and the probability of character occurrence, define \( A_1 \), and then make:

\[
\bar{l} = \left[ \log_2 \frac{1}{P(a_i)} \right]
\]  
(14)

\[
\log_2 \frac{1}{P(a_i)} \leq \bar{l} \leq \log_2 \frac{1}{P(a_i)} + 1
\]  
(15)

From the above equation (25), it can be concluded that: \( 2^{-\bar{l}} \leq P(a_i) \), Therefore:

\[
\sum_{i=1}^{N} 2^{-\bar{l}} \leq \sum_{i=1}^{N} P(a_i) = 1
\]  
(16)

If \( n \) is an arbitrary integer, then:

\[
\left[ \sum_{i=1}^{N} 2^{-l_i} \right]^n = \left( \sum_{i=1}^{N} 2^{-l_i} \right) \left( \sum_{i=1}^{N} 2^{-l_i} \right) \cdots \left( \sum_{i=1}^{N} 2^{-l_i} \right)
\]  
(17)
Index $l_1 + l_2 + \ldots + l_n$ is the code length of $n$ codes, which the minimum length should be greater than or equal to $n$.

According to the theorem, given a set of integers, the following (equation 17) is satisfied:

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1.$$  

Prefix codes with codeword length $l_1, l_2, \ldots, l_N$ can always be found. Without losing generality, suppose $l_1 \leq l_2 \leq \ldots \leq l_N$, and define the weight value $w_1, w_2, \ldots, w_N$ as: $w_1 = 0$, $w_j = \sum_{i=1}^{j-1} 2^{-l_i}$, $j > 1$. For $j > 1$, the binary representation of $w_j$ will occupy $\lceil \log(w_j + 1) \rceil$ bits.

Therefore, the length of prefix code is:

$$\log_2 (w_j + 1) = \log_2 \left[ \sum_{i=1}^{j} 2^{l_i - j} + 1 \right]$$
$$= \log_2 \left[ 2^{j} \sum_{i=1}^{j} 2^{-l_i} + 2^{-j} \right]$$
$$= l_j + \log_2 \left[ \sum_{i=1}^{j} 2^{-l_i} \right]$$

$$\leq l_j$$

Therefore, the length of the improved adaptive Huffman variable length coding is as follows:

$$H(m) \leq l \leq H(m) + 1 \quad (19)$$

Given the word length $m$, MBD model data sequence $x = (x_1, x_2, \ldots, x_m)$, map $[0, 1]$ to the $2^m$ binary range, and define $n_j$ as the number of times that symbol $j$ appears in a sequence of length $\text{Total count}$, then $F_j(k)$ can be estimated as follows:

$$F_j(k) = \frac{\sum_{i=1}^{n_j} n_j}{\text{Total count}} \quad (20)$$

If defined

$$I^{(e)} = I^{(n-1)} + (u^{(e-1)} - I^{(e-1)})F_x(x_n - 1) \quad (21)$$
$$u^{(e)} = u^{(e-1)} + (u^{(n-1)} - I^{(e-1)})F_x(x_n) \quad (22)$$

$I^{(e)}$ and $u^{(e)}$ are the starting points of coding. The code is:

$$l = l + \frac{(u-l+1) \times \text{Cum count}(x-1)}{\text{Total count}} \quad (23)$$
$$u = l + \frac{(u-l+1) \times \text{Cum count}(x)}{\text{Total count}} - 1 \quad (24)$$

By converting $u$ and $l$ into binary numbers, the node code of the tree can be obtained. The left subtree is coded as "0" and the right subtree is coded as "1". When encountering a node that stores characters, output the code of the character, and encode the nodes of the tree in turn until all nodes are encoded to achieve the purpose of data compression.
3. Result
In this paper, MBD model of automatic packing machine is used as the experimental object, to obtain the data information of the model and the MBD model is tested with different compression algorithms. The results are shown in table 1.

| Algorithm                      | Before compression/bit | After compression/bit | Compression ratio | Compression time /s |
|--------------------------------|------------------------|----------------------|-------------------|---------------------|
| Huffman compression algorithm  | 3688896                | 1232581              | 2.99              | 3.69                |
| Greedy Huffman compression algorithm | 3688896                | 1232581              | 3.06              | 3.41                |
| Adaptive Huffman compression algorithm | 3688896                | 1232581              | 2.18              | 1.21                |
| Improved adaptive Huffman compression algorithm | 3688896                | 1232581              | 3.87              | 1.11                |

The analysis and comparison results from the above table is as follows: in the same data of MBD model, the compression rate from large to small is: improved adaptive Huffman compression algorithm, greedy-Huffman compression algorithm, Huffman compression algorithm, adaptive Huffman compression algorithm; the compression time of the four algorithms is from fast to slow: the improved adaptive Huffman compression algorithm, adaptive Huffman compression algorithm, greedy-Huffman compression algorithm, Huffman compression algorithm. By the results, the improved adaptive Huffman compression algorithm is better for data lightweight of the MBD model. In the relatively strong computer hardware equipment, the improved adaptive Huffman compression algorithm has higher transmission efficiency and compression rate, while can better solve the problem of large memory space occupied. The feasibility and correctness of the lightweight method are verified.

4. Conclusion
Additive manufacturing has become the research focus in the field of manufacturing industry, which can quickly realize digital model entity manufacturing. The products produced by additive manufacturing technology have good performance and mechanical properties. In order to solve the problem of slow remote transmission of 3D model used in additive manufacturing, a data lightweight method for MBD model is proposed in this paper, which makes the data file smaller and faster, without losing the model information. Since this method solves the problem on the data transmission and acquisition, it provides a solution on the 3D model remote transmission for online additive manufacturing.

ACKNOWLEDGMENT
This paper is supported by GDAS' Project of Science and Technology Development (2017GDASCX-0115), GDAS' Project of Science and Technology Development (2018GDASCX-0115), and GDAS'Special action for building a first-class research institution in China (2020GDASYL-20200402007).

REFERENCES
[1] Z.G. Wu, H.X. Chen, Research on lightweight engineering application of MBD digital model for complex shell. Innovation and application of science and technology, vol. 16, pp. 37-38, 2017.
[2] Q. Virqilio, R. Louis, P. Robert, Will model-based definition replace engineering drawings throughout the product lifecycle. A global perspective from aerospace industry. Computers in industry, vol. 61, pp. 497-508, 2010.

[3] H. Lu, Y.Q. Fan, The Study of the Drawing Functions in the Digital Aero Product Definition. Journal of engineering graphics, vol. 2, pp. 29-34, 2008.

[4] B.H. Hu, L.B. Wen, MBD-Based Three-Dimensional Digital Assembly Process Design and on-Site Visualization Technology Application. Assembly process, vol. 22, pp. 81-85, 2011.

[5] W.Z. Wang, Research and Implementation of 3D model lightweight technology. Huazhong University of science and technology, Wu han, 2015.

[6] G. Taubin, J. Rossignac, Geometric compression through topological surgery. ACM Transactions on Graphics, 17th ed., vol. 2, pp. 84-115, 1998.

[7] C. Ouma, C. Gotsman, Triangle mesh compression, Proceedings of the GraphicsInterface, Vancouver, 1998, pp. 26-34.

[8] W. D. Li, S. K. Ong, J. Y. H. Fuh, Feature-based design in a distributed and collaborative environment. Computer-Aided Design, 36th ed., vol. 9, pp. 775-797, 2004.

[9] Q.F. Wang, L. Yang, Y.B. Huang, Model Simplification in Collaborative Design. Journal of computer-aided design & computer graphics, vol. 01, pp. 108-113, 2006.

[10] S. CE, A mathematical theory of communication. The Bell System Technical Journal, 27th ed., vol. 7, pp. 379-423, 1948.

[11] Wang, P., Mao, Z. M. (2001) Realization and Research of LZSS Algorithm for Chinese Text. Computer Engineering, 27(7):379-423.

[12] C.F. Zheng, Research of Several Common Lossless Data Compression Algorithms. Computer technology and development, 21th ed., vol. 9, pp. 73-76, 2011.

[13] D.Q. Wang, Study on Data Compression for Fault Recorder in Power System. Southwest Jiaotong University, Chengdu, 2003.