A New Parallel-by-Cell Approach to Undistorted Data Compression Based on Cellular Automaton and Genetic Algorithm

GU Jing (顾 静) and SHUAI Dianxun (帅典勋)†
State Key Laboratory of Intelligence Technology and System
Tsinghua University, Beijing 100084, P.R. China
†Department of Computer Science and Technology
East China University of Science and Technology, Shanghai 200237, P.R. China

Received August 10, 1998; revised November 23, 1998.

Abstract In this paper, a new parallel-by-cell approach to the undistorted data compression based on cellular automaton and genetic algorithm is presented. The local compression rules in a cellular automaton are obtained by using a genetic evolutionary algorithm. The correctness of the hyper-parallel compression, the time complexity, and the relevant symbolic dynamic behaviour are discussed. In comparison with other traditional sequential or small-scale parallel methods for undistorted data compression, the proposed approach shows much higher real-time performance, better suitability and feasibility for the systolic hardware implementation.

Keywords data compression, genetic algorithm, cellular automaton, parallel processing

1 Introduction

A versatile real-time approach to data compression is essential to many areas, such as network communication, multimedia, database, knowledge engineering, information system, image processing and so forth. There are mainly two paradigms of data compression: reversible undistorted noise-free coding and irreversible distorted entropy-based coding. Although neural networks have been widely applied to the distorted compression, they are still open to question to be used in the hyper-distributed hyper-parallel reversible compression. On the other hand, however, conventional sequential or small-scale parallel methods for data compression encounter many formidable problems in real-time performance and systolic hardware implementation. Moreover, the difference among the present distinct methods of undistorted compression with respect to algorithms, data structures and hardware schemes results in application complication and poor universality.

The genetic algorithm (GA) has been successful in optimization problem solving, dynamic system modeling, automatic programming, intelligent controlling, robot planning, immunity system and financial system analyzing[1-8]. As for distorted data compression, there is probably no substantial difficulty in using GA to obtain a feasible suboptimal solution. Nevertheless, to our knowledge, a lot of work remains to be done as for reversible undistorted compression by using GA as a universal approach to various data and different compression demands.

This paper proposes a new parallel-by-cell approach to undistorted data compression, which uses a genetic evolutionary algorithm to obtain the local rules for a data-compressing cellular automaton. The proposed approach is suitable for many kinds of data types and many different compression requirements, and has advantages over other conventional approaches with respect to higher real-time performance and much simpler systolic structure.
for hardware implementation. An important issue that attention should be paid to when using an automaton and a genetic algorithm is how to deal with more and more shrunk length of the data during compressing process through a constant length of chromosomes used in a genetic algorithm and a constant size of an automaton. This paper describes the basic principle and the correctness of the cellular parallel data compression, and discusses its time complexity, the relevant symbolic dynamic behaviour and some comparisons with other traditional methods for undistorted compression.

2 Cellular Automaton for Data Compression

Consider a cellular automaton composed of cells evenly lying on a half line, a positive integer \( i \leq N \) and a symbol in a finite set \( S \) representing the coordinate of a cell and the state of a cell, respectively. A string of symbols over \( S \) is called a configuration of the automaton.

**Definition 1.** A one-dimension cellular automaton of data compression is a quintuple \( G = (S, \sigma, F, X_0, R) \), where \( S \) is a finite set of symbols; \( \sigma \) is a cyclic shift operator over \( S^N \), left shifting a string a position once and shifting the symbol at the most left cell into the most right cell, denoted as \( \sigma'(X) = \sigma^{-1}(\sigma(X)) = X \in S^N \); \( F \) is a continuous mapping exchangeable with respect to \( \sigma \), i.e. \( F : S^N \rightarrow S^N \), \( F(\sigma^k(X)) = \sigma^k(F(X)) \), for \( \forall X \in S^N \) and any non-negative integer \( k \); \( X_0 \) is a given initial configuration, \( X_0 \in S^N \); and \( R \) is an equivalent relation partitioning the coordinates of a given configuration.

**Lemma 1.** For one-dimension data-compressing cellular automaton \( G = (S, \sigma, F, X_0, R) \), there are a neighbour radius \( r \) and a local mapping \( f : S^{2r+1} \rightarrow S \) independent of the coordinates of cells, so that \( F(X) = [f(x(i+N--r)\mod N,...,x_i,...,x(i+r)\mod N)] \mid i \in N^N \) holds true for any \( X = x_1 ... x_N \in S^N \), where \( [f(\cdots)]^N \) means that the mapping \( f \) is concurrently executed for all the cells.

**Proof.** Let \( F(X) = Y = y_1 ... y_N \in S^N \), \( X = x_1 ... x_N \in S^N \), \( x_i, y_i \in S \) for \( i \in \{1,...,N\} \). Assume that \( x_i = x_j, i < j \), and there is no symbol equal to \( x_i \) between \( x_{i+1} \sim x_{j-1} \). If for any radius \( k \) there are \( x(i+N-k)\mod N = x(j+N-k)\mod N \) and \( x(i+k)\mod N = x(j+k)\mod N \), then \( X \) is the string with the minimal cycle \( x_i ... x_j \), so that \( \sigma^{(j-i)}(X) = X \), \( F(\sigma^{(j-i)}(X)) = F(X) = Y \). Therefore, \( F(\sigma^{(j-i)}(X)) = \sigma^{(j-i)}(F(X)) = \sigma^{(j-i)}(Y) = Y \), \( y_i = y_j \). Similarly, for any \( n \geq 0 \), there exists \( F(\sigma^{(j-i)}(X)) = \sigma^{(j-i)}(Y) = Y \). It implies that through the mapping \( F \), the same symbol \( x \) at different coordinates of a configuration \( X \) is mapped into the same symbol \( y \) at the corresponding coordinates of \( Y = F(X) \). The map \( Y \) of \( X \) with cycle \( x_1 ... x_{j-1} \) must have the cycle \( y_1 ... y_{j-1} \). As a result, constructing the following local mapping:

\[
\begin{align*}
&f(x(i+N-r)\mod N,...,x_i,...,x(i+r)\mod N) = y_i, \\
&\text{with neighbour radius } 1 \leq r_i \leq i-j,
\end{align*}
\]

can yield the same \( Y \) as \( F(X) \). Further, consider the case that there is a minimal neighbour radius \( k_i \) that \( x(i+N-k_i)\mod N \neq x(j+N-k_i)\mod N \).

If \( y_i = y_j \), then the neighbour radius \( r_i \geq 1 \) is allowed. If \( y_i \neq y_j \), then an allowable radius is \( r_i \geq k_i \) because the local mappings:

\[
\begin{align*}
&f(x(i+N-r)\mod N,...,x_i,...,x(i+r)\mod N) = y_i, \\
&f(x(j+N-r)\mod N,...,x_i,...,x(j+r)\mod N) = y_j
\end{align*}
\]

become a single-valued mapping of \( S^{2r+1} \rightarrow S \) with respect to coordinates \( i \) and \( j \). Consequently, for each pair, \( x_i = x_j \), an allowable minimal neighbour radius \( r_i \) can be found, denoted as \( (x_i = x_j) \Rightarrow r_i \). Given configuration \( X \) and taking \( r_m = \max\{r_i \mid x_i = x_j \Rightarrow r_i \} \) as the minimal neighbour radius, it is obvious that the single-valued local mapping \( f \) satisfying \( F(x(i+N-r)\mod N,...,x_i,...,x(i+r)\mod N) \mid i \in N^N \) can be successfully constructed. \( \square \)

**Definition 2.** A text-compressing cellular automaton \( G = (S, \sigma, F, X_0, R) \) puts into effect the text data compression as follows: for any \( X_0 \in S^N \), \( \lim_{n \to \infty} F^n(X_0) = X_f \), where \( F \)