Information Sciences 439–440 (2018) 39–49

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Efficient chain code compression with interpolative coding

Borut Žalik[1], Domen Mongus, Niko Lukač, Krista Rizman Žalik*

Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, Maribor, SI-2000, Slovenia

ARTICLE INFO

Article history:
Received 2 June 2017
Revised 15 January 2018
Accepted 25 January 2018

Keywords:
Chain codes
Lossless compression
FELICS coding
Ψ-coding

ABSTRACT

This paper considers the use of interpolative coding for lossless chain code compression. The most popular chain codes are used, including Freeman chain code in eight (F8) and four directions (F4), Vertex Chain Code (VCC), and three-orthogonal chain code (3OT). The whole compression pipeline consists of the Burrows–Wheeler transform, Move-To-Front transform and the interpolative coding, which was improved by FELICS and new Ψ-coding. The approach was compared with the state-of-the-art chain code compression algorithms. For VCC, 3OT and F4, the obtained results are slightly better than the existing approaches. However, an important improvement was achieved with F8 chain code, where the presented approach is considerably better.

1. Introduction

Visual perception provides humans with the crucial information of their surroundings [10], which is the reason why computer scientists have tried to provide users with graphical output since the early days of computing [12]. For technical reasons, graphical information is, today, displayed exclusively on raster devices, and therefore, various rasterization algorithms have been proposed for straight lines [6], polygons [34], and curves [24]. Unfortunately, rasterized space requires a considerable amount of memory. For this reason, various image compression techniques have been developed [36]. However, in some cases, the users are interested only in individual shapes rather than the entire raster space. In such cases, chain codes can be applied as an efficient alternative. A chain code stands for a sequence of instructions which control the walk through the border pixels of a considered shape. Various chain codes were developed, starting with the Freeman chain code in eight (F8) and four (F4) directions [13], Vertex Chain Code (VCC) [7], Three OrThogonal chain code (3OT) [42], and Unsigned Manhattan Chain Code (UMCC) [46]. Illustrative chain code explanations can be found in the literature (e.g. [22,25,40]). Chain codes also have, besides the compact shape's representation, many other useful properties and, therefore, have a wide range of applications, such as, for example, in pattern recognition and image processing [21,44], font representation in embedded systems [14], or geographical information systems [3]. 3D chain codes are also known for 3D voxelized shapes [8,19,41].

Despite the fact that the chain codes already describe the border of rasterized shapes efficiently, they can be compressed further. A new chain code compression method is introduced in this paper. The method consists of three steps: Burrows–Wheeler transform [1] is applied firstly, Move-To-Front transform [35] follows, and the interpolative coding is the last [28]. This combination has not yet been used for chain code compression. Furthermore, the coding of the considered interpolative

* Corresponding author.
E-mail addresses: borut.zalik@um.si (B. Žalik), krista.zalik@um.si (K.R. Žalik).

[1] http://gemma.feri.um.si/

https://doi.org/10.1016/j.ins.2018.01.045

coding element was enhanced in our approach, either with the FELICS code, or with the new Ψ codes. As confirmed by experiments, the proposed approach is an attractive and easy to implement possibility for lossless chain code compression, which is better than the state-of-the art.

The paper consists of five Sections. The previous works for chain code compression are considered in Section 2. The proposed approach is described in Section 3. Section 4 introduces the results of experiments. Section 5 concludes the paper.

2. Previous works

Kaneko and Okudaira [16] experimented with chain codes while compressing curves obtained from geographic maps. They exposed long segments with gentle curvature. Such segments can be represented with only two symbols. They showed by experiments that their approach outperforms the classical F8 coding by 50–60%. Directional Difference Chain Code (DDCC), derived from the Freeman chain code in eight directions (F8), was presented in [22]. The authors considered more than 1000 shapes to obtain the statistical information on how the F8 chain code symbols change from one to another. The Huffman chain codes were determined for each change based on the statistics. The obtained compression ratio was better than 1.5. A similar approach was proposed in [38], where it is referred to as the Angle Freeman chain code of eight directions (AF8). The Modified Directional Freeman chain code in eight directions was proposed in [37]. It explores the pieces of discrete straight lines and, together with the probability of chain code symbols in the contours, proposed the nine-symbols code (MDF9). The authors confirmed by experiments that the resulting code was more efficient than the referenced ones. Three simple approaches for VCC compression (V_VCC, E_VCC, and C_VCC) were suggested in [25]. The alphabet Σ_{VCC} consists of only three symbols, i.e. $\Sigma_{VCC} = \{1, 2, 3\}$, which are normally coded with two bits. As symbol 2 is more probable, it can be coded with one bit (e.g. 0), while the other two symbols with two bits (e.g. 10 and 11). The authors considered this approach as a variable V_VCC. The extended VCC (E_VCC) applies unused bits for the most frequent combination of symbols 1 and 3. The compressed VCC, C_VCC, uses Huffman codes for VCC codes and some of their combinations as follows (code 2 is coded with bit 0, code 1 with 1110, code 2 with 1111, code combination 1–3 with 10, and code combinations 3–1 with 110). In general, C_VCC performs the best, while E_VCC is the worst. The improved versions, named Dynamic VCC (D_VCC) and Equal-Length Compressed VCC (EC_VCC), with considerably better compression performances, were proposed in [20]. A Modified Three Orthogonal chain code (M_3OT) was suggested in [39], where an additional three symbols were used to code the frequent combinations of 3OT symbols 0 and 1. The arithmetic coding was applied in the final step. The suitability of the arithmetic coding against static Huffman coding was confirmed in [2] when compressing three-bit chain codes. The first and the second-order Markov models for Directional Difference Freeman chain code in eight directions was used in [26]. A statistical model was built for coding the next chain code element based on the context of the previous one or two codes. A context tree was used to predict the next chain code symbol based on previously seen codes [3]. The prediction was then coded by the arithmetic coding. The conditional Markov model was used in [4], where the selection between different Markov orders is done by the Bayesian information criterion. The model with the minimal criterion is selected and used by the arithmetic coding. Markov models have been used up to the order 5. Directional Grid Chain Coding (DGCC) was introduced in [33], where the coding efficiency is realised as a Markov chain of neighbouring relations. The authors developed lossless and quasi-lossless variants. Move-To-Front (MTF) transform was used on VCC, 3OT and NAD chain codes in [45]. The sequence of MTFs was applied, and those which produced the sequence of symbols with the smallest information entropy was selected. The RLE, followed by the VLC, were then applied for the compression. In general, less than one bit per chain code is needed. A chain code independent algorithm for chain code compression was proposed in [47]. At first, chain code symbols are binarized and, after that, they are compressed by the combination of RLE and a variation of LZ77. The method was tested on F8, F4, VCC, 3OT and NAD chain codes. Recently, the study was performed of the most suitable combination of string transformation techniques for chain code compression [48]. Various combinations of Burrows–Wheeler Transform, Move-To-Front transform and zero-Run-Length were tried for each of the chain codes. Based on experiments, the guidance was given for the best combination for each chain code. RLE, with VLC or PAQ8L, was used for the final redundancy removal.

In the majority of cases, chain code compression algorithms are lossless. However, a few quasi-lossless approaches also exist [23,31,33].

3. Chain code compression with interpolative coding

A method for chain code compression, based on interpolative coding, is introduced in the continuation. Let $S_C = \{s\}$ be the input sequence of $n = |S_C|$ chain code symbols $s \in \Sigma_C$, where Σ_C denotes the alphabet of the considered chain code, $C = \{F8, F4, VCC, 3OT\}$ in our case. The algorithm consists of the following steps:

1. S_C is firstly transformed with the Burrows–Wheeler's Transformation (BWT). The new symbols' sequence $SB_C = BWT(S_C)$ is obtained.
2. SB_C is then transformed with Move-To-Front (MTF) transform to the sequence $SM_C = MTF(SB_C)$. SM_C contains, hopefully, long sequences of 0-symbols.
3. SM_C is turned to the strictly increasing sequence of integers SN_C.