Lossless Multi-Grid Chain Code with Optimized Cell Usage for Contour Description of Visual Objects

Fabiana O. P. Fernandes, Willian G. de Almeida and Flávia M. Freitas
Programa de Pós-graduação em Engenharia Elétrica e Curso de Engenharia Eletrônica e de Telecomunicação
PUC Minas
Belo Horizonte, MG, Brazil
fabianapaixao@ieee.org

Abraham Alcaim
CETUC – Centro de Estudos em Telecomunicações
Programa de Pós-graduação em Engenharia Elétrica
PUC Rio
Rio de Janeiro, RJ, Brazil
alcaim@cetuc.puc-rio.br

***Abstract*—In the context of object-based image compression, shape encoders that describe the contour of the objects, called chain codes, tend to be more efficient than that the one based on bitmap, which is the MPEG-4 Part 2 framework. Among the chain codes, those based on grid cells are expected to outperform the ones that employ a symbol to code each particular boundary pixel. This paper presents a lossless chain code based approach for binary shapes encoding, called OCU-LMGCC (Optimized Cell Usage - Lossless Multi Grid Chain Code). The OCU-LMGCC solves shortcomings in the original MGCC, optimizes the required extra parallel information to the decoder and uses a new set of coding symbols, better suited to the novel cell-switching schemes proposed. Results showed a compression gain noticeable larger than the known algorithms reported in the literature.**

Keywords-shape encoding, object-based image compression, MPEG-4, chain code, grid cell.

I. Introduction

Recent video coding research gave birth to the latest video coding standard H.264/AVC (MPEG-4 Part 10), whose compression performance significantly exceeds previous standards. However, only MPEG-4 Part 2 provides object-based compression, with novel capability of independently handling and jointly encoding arbitrarily shaped objects. As new multimedia formats emerge, object-based encoding intends for scalability and interactivity, storage and retrieval of object-based data, representation and manipulation of objects in a scene in postproduction of TV and cinema, mixture of synchronized natural and synthetic objects in computer games and mobile multimedia applications [1][2].

The shape compression in MPEG-4 Part 2 framework uses a binary alpha plane to inform the region of support of a specific binary visual object. However, the chain codes, which provide a contour description based on edges of objects of arbitrary shape, tend to be less redundant in terms of compression. The main contribution of this work is focused on the proposition of an arbitrary-shape lossless approach based on chain code using grid cells to describe the object contour. The proposed algorithm is called OCU-LMGCC (Optimized Cell Usage - Lossless Multi Grid Chain Code). Firstly, it solves shortcomings of the original lossless and quasi-lossless MGCC (Multi Grid Chain Code) [7]. It also presents contributions aiming at optimizing the required parallel information for lossless shape compression. At last, it applies novel cell-switching schemes, combined with a better suited set of coding symbols. All these contributions when putting together enable to minimize the coding symbols necessary to describe the contours.

This paper is organized as follows: Section II briefly describes the main chain coders known in the literature. Section III highlights the fundamental contributions of the OCU-LMGCC, particularly related to the lossless MGCC [7][8]. Results for binary alpha plane lossless coding of MPEG-4 video test images using the OCU-LMGCC here proposed and other chain coders [5][6][7][8] are presented and compared to MPEG-4 framework published results [7], in Section IV. Finally, in Section V, the main conclusions of this work are found.

II. Previous Works

We propose to classify the chain coders into two groups: (i) Neighborhood-based encoders, which generate a symbol for each boundary pixel (among them, stand out the first Chain Code (CC) [4], the Differential Chain Code (DCC) [5] and the Conditional Differential Chain Code (CDCC) [6]; (ii) Cell-based encoders, that utilize grid cells which can represent a set of boundary pixels. The Multi-Grid Chain Code (MGCC) [7] and the Directional Grid Chain Code (DGCC) [8] are found in this group. For all the chain coders, both for the first and for the second group, the absolute position of the first boundary pixel of the object must be encoded as parallel information.

A. Neighborhood-Based Encoders

The first proposed chain code (CC) [4] encodes d_i, which is the relative position between the current boundary pixel and its antecessor. The algorithm presents two variations, CC8 and CC4, using a neighborhood of eight or four pixels, respectively. The Differential Chain Code (DCC) [5] encodes the differential direction k_i, which is the difference between d_i and d_{i-1} (related to the current and to the previous boundary pixel, respectively). In turn, the Conditional Differential Chain Code (CDCC) [6] estimates the current k_i and encodes its prediction error e_i. The estimate chooses the symbol with maximum conditional probability and probabilities are determined in advance from a large set of objects which have their contour encoded.

B. *Cell-Based Encoders*

The Multi Grid Chain Code (MGCC) [7] uses cells corresponding to a square area of 3x3 pixels to scan the contour of the object and contour tracking is done clockwise. It uses two types of cells, clockwise and counterclockwise, as shown in Fig. 1. The cells usually encode more than a simple boundary pixel, especially for smooth contours.

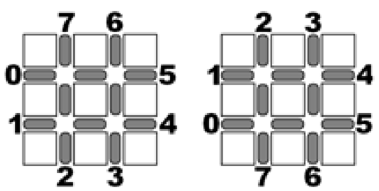

Figure 1. Type of cells in MGCC: (a) counterclockwise; (b) clockwise.

The positioning of the cell over the contour moves dynamically. The entry segment into the cell is positioned so that the first boundary pixel is below the element ‘0’, as shown in Fig. 2, in which a clockwise cell is employed. The remaining segments of the cell (1, 2, …, 7) are exit segments and the entry element of the next cell must match the exit element of the current one. Accordingly to [7], once the first boundary pixel is positioned below the segment ‘0’, the next boundary pixel within the cell is searched in the following order of priority: “right”, “straight ahead” and “left”. This procedure is repeated until there is no possibility of movement inside the cell. The complete coding of the contour ends when a boundary pixel already encoded is achieved again. Although [7] reports the use of a cell-switching scheme that codes as many large movements as possible, the scheme itself has not been described. Only later in [8], a scheme the makes the exchange of the future cell whenever the exit of current cell is either ‘1’, ‘2’ or ‘3’, was presented.

The use of a 3x3 cell implies in a small ambiguity while decoding the contour path inside a cell, since there is no way to distinguish whether its central pixel is or not a boundary pixel. In this case, a quasi-lossless shape encoder is provided. This ambiguity was solved in [7] through the increment of 1 bit per cell as extra parallel information, so that such distinction is made by the encoder and informed to the decoder. In this case, a lossless shape encoding is achieved. Results in [7] show that lossless MGCC hardly presents better performance than DCC [5]. Hence it suggests DCC for lossless encoding and MGCC for quasi-lossless encoding.

The Directional Grid Chain Code (DGCC) [8] uses rectangular cells of 2x3 pixels in order to avoid ambiguity relative to the central pixel. The DGCC uses eight different types of cell, aiming at providing adaptability to trends of the contour direction. The types depend on “guidance” (referring to the lengthening of the cell - horizontal or vertical), on “entry” (referring to the entry position of the cell - by the side or by the base) and finally, on “direction” (indicating the direction of the contour trajectory inside the cell - to the right or to the left). The determination of the next cell depends on the type of the current cell and its associated exit. It is made using an extensive table of logic tests that aims at increasing the probability that a larger portion of contour is covered by the next cell. However, the large number of cell types and the switching scheme among them can impose some constraints on implementing the DGCC.

III. Optimized Cell Usage - Lossless Multi Grid Chain Code (OCU-LMGCC)

The OCU-LMGCC focuses on solving shortcomings of the original MGCC [7] and provides other improvements in order to increase code performance for lossless shape encoding, as related in the following.

A. *Correction of Shortcomings of MGCC*

1) Searching Rule for the Next Boundary Pixel: If the searching rule presented in [7] is applied, the cell moves into the interior of the object, rather than it scans its contour. In fact, this rule must be corrected in such a way that the next boundary pixel in the cell is searched in the following priority order: “left”, “straight ahead” and “right”. An example of using a clockwise cell to scan a contour’s section using the new searching rule is shown in Fig. 2, in which boundary pixels are highlighted in black.

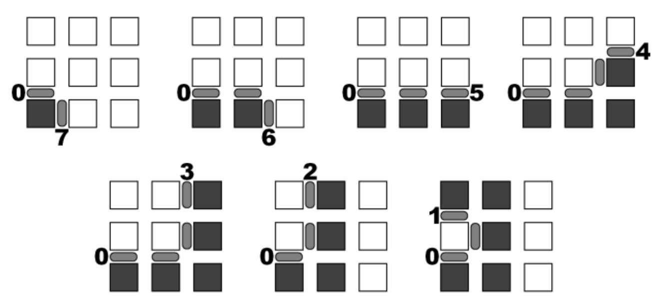

Figure 2. Section of the contour scanned by a clockwise cell.

2) Exceptional Event Coding (Failure of the Searching Rule): Another shortcoming of the MGCC [7] is that it lacks the procedure to describe the situation at which the current pixel is the central one and if the established searching rule for the next pixel is used, no other pixel is found. This causes the interruption of the contour tracking. In clockwise cell, for example, this occurs at situations exemplified in Fig. 3a and 3b, in which the first pixel to be scanned by the cell is A. The searching rule then finds the next pixel B “straight ahead”.

Following, pixel C is found “on the left”. However, when C is the current pixel, if the searching rule is applied, no other pixel can be found, since pixel B, already scanned, is located “backward” in relation to C.

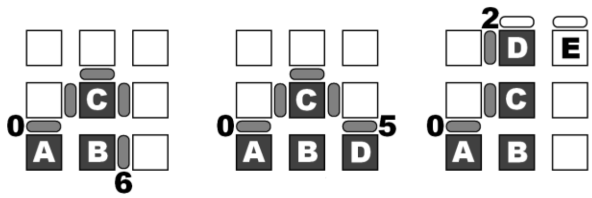

Figure 3. Exceptional evevent conding:(a) Situation described by symbol R; (b) Situation described by symbol R2; (c) Example of invalid path that must be avoided.

To overcome this constraint, the OCU-LMGCC incorporates two new symbols - R_1 and R_2 - to the original set of exits, in order to code each of both situations. Thus, the symbol set of the OCU-LMGCC becomes (1, 2, …, 7, R_1, R_2). Symbol R_1, when emitted by a clockwise cell, will mean that pixels A, B and C have been scanned and that the exit of the cell occurs by element ‘6’. Differently, symbol R_2 will code pixels A, B, C and D, signaling the exit occurring through element ‘6’. Equivalent situations are informed for counterclockwise cell.

3) Invalid Path at Cell’s Border (Exception to the Searching Rule): In MGCC [7], the occurrence of invalid paths in the cell is not foreseen. However, Fig. 3c presents a situation at which the application of the searching rule if D is the current pixel will achieve the pixel E. However, if pixel E is encoded by the same cell, a path of contour located at the border of the cell will be coded, what is not allowed (this invalid path is represented by cones not filled by black). Therefore, the correct solution is to assume that the exit of the cell occurs by element ‘2’, leaving the pixel E to be encoded by the next cell. The OCU-LMGCC provides the analysis of all these prohibited conditions. In the occurrence of them, it interrupts the application of the searching rule and informs the correct exit in order to avoid prohibited contour paths.

B. Other Improvements upon Lossless MGCC

1) Optimization of the Extra Parallel Information to Undo Ambiguity Related to the Central Pixel: The lossless MGCC [7] requires to send one extra bit per cell in order to inform the decoder whether the central pixel belongs to the object contour. In OCU-LMGCC, this parallel information is sent only if it is strictly necessary, since there are some situations at which ambiguity definitely does not exist. Fig. 4 shows an example where the extra bit is required, corresponding to the exit of the clockwise cell through segment ‘2’. However, still in clockwise cell, if the emitted symbol is either ‘5’, ‘6’, ‘7’, ‘R_1’ or ‘R_2’, and in counterclockwise cell, if the sent symbol is either ‘1’, ‘2’, ‘R_1’ or ‘R_2’, this extra bit is unnecessary. Explanations for the case of clockwise cell are given in the following:

- Outputs ‘R_1’ and ‘R_2’: the central pixel is obligatorily a boundary pixel, as shown in Fig. 3a and 3b;
- Outputs ‘5’ and ‘6’: the central pixel does not belong to the contour, otherwise the ‘rare events’ ‘R_1’ and ‘R_2’ would have been emitted, instead;
- Output ‘7’: the central pixel cannot belong to the contour if neither its left neighbor nor its downstairs neighbor belong;

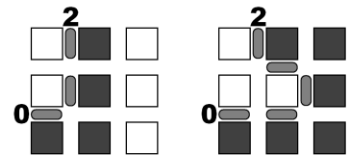

Figure 4. Example of ambiguity with respect to the central pixel.

2) New Switching Schemes for the OCU-LMGCC: The switching scheme proposed for MGCC [8] is run by encoder and decoder and it is supposed to determine the best next cell to be applied, depending on the current cell type and its associated exit. However, we verified through experimental tests that schemes which promote the determination of the current cell itself (rather than the next cell) are more efficient in terms of compression, even considering the disadvantage of requiring one additional bit per cell to report to the decoder the type of the employed cell by the encoder. Thus we propose two novel schemes:

- Observation-Based Scheme: designed from our empirical observations about what happens when the exit of the current cell is either ‘6’ or ‘7’. If the current cell is commuted, more pixels can be encoded most often.
- Pixel-Based Scheme: computes the amount of boundary pixels covered by each type of cell and chooses as current cell the one which codes the largest number of pixels.

3) Extended Symbols Set: The novel switching schemes here proposed require to inform the decoder about the cell type. A rudimentary implementation would point out the need for transmitting an additional bit per cell. Alternatively, the OCU-LMGCC proposes to use a symbol set $S = \{(x,y)\}$ with 18 elements, corresponding to the combination of each cell type x with all nine possible exits *y*. The code entropy achieved with this proceeding is smaller than that obtained when the encoder employs the nine symbols plus an additional bit per cell.

IV. Simulation Results

Table I presents results relative to the coding of the alpha planes of the following set of MPEG-4 images: ‘Lena’, object 2 of frame 1 of sequence ‘Weather’, objects 1, 2 and 3 of image ‘Pepper’, object 1 of frame 1 of sequence ‘Children’ and finally, objects 1, 2 and 3 of frame 10 of sequence ‘Fish and Logo’. The table compares the OCU-LMGCC to the chain coders of the first group (DCC8 and CDCC8) and of the second group (lossless MGCC associated with MGCC-scheme presented in [8] and finally, DGCC). The achieved bit rate *R* after Huffmann coding is given in bpbp (average bits per boundary pixel). The gain *G* refers to the reduction in bit rate achieved by the best variation of the OCU-MGCC, in comparison with other coders.

TABLE I. AVERAGE RESULTS OF CODING PERFORMANCE

Encoder	Algorithm's variation	R (bpbp)	G
DCC8 [5]	-	1,6556	27,3%
CDCC8 [6]	-	1,6095	25,3%
MGCC[7]	shortcomings corrected and MGCC-Scheme [8]	1,778	32,3%
OCU-MGCC	***Observation-Based Scheme***		
	1 bit/cell if necessary	1,5305	21,4%
	1 bit/cell always	1,6094	25,3%
	Pixel-Based Scheme		
	1 bit/cell if necessary	1,2029	0,0%
	1 bit/cell always	1,384	13,1%
DGCC	-	1,183	-1,7%

Results relative to lossless MGCC [7] in Table I were obtained after the correction of the shortcomings reported in Secion III-A. The implementation of lossless MGCC also adopted: (i) the MGCC-Scheme [8] to determine the next cell, (ii) the use of (1, 2, …, 7) as the set of coding symbols and (iii) one extra bit per cell to undo ambiguity related to central pixel.

Instead, the implementation of the OCU-LMGCC in association to the novel switching schemes reported in Section III-B-2 employed the set of 18 symbols as proposed in Section III-B.3. The impact of optimizing the parallel information, as related in Section III-B-1, can also be verified.

In OCU-LMGCC, the Pixel-Based Scheme achieved a bit rate 14% and 21.4%, respectively, less than the correspondent algorithm’s variations using the Observation-Based Scheme. Also, from Table I it can be inferred an average gain of 5% (for Observation-Based Scheme) and 13% (for Pixel- Based Scheme) at compression efficiency when the parallel information needed to undo ambiguity is optimized. When the coding performance of the best variant of OCU-LMGCC is compared to the bit rate achieved by DGCC, this last outperforms the first at only 1.7% (in contrast, OCU-LMGCC is much simpler than DGCC). But when the best variant of OCU-LMGCC is compared to the original lossless MGCC [7], the first performs 32.3% better. It also outperforms DCC8 [5] at 27.3% (note that DCC8 has been appointed in [7] as the better option for lossless encoding).

An important remark is that cell-based chain coders usually have to code more boundary pixels than the neighborhood-based chain coders and that is the reason for better performances of DCC8 and CDCC8 in relation to the original MGCC. This is due to the neighborhood of eight pixels used by DCC8 and CDCC8, which allows the coding of consecutive boundary pixels located 45, 135, 225 or 315 degrees relative to each other (in additional to a boundary pixel only located on the left, on the right, straigth above or backwards of its antecessor). However, due to particularities of the algorithms of OCU-MGCC and DGCC, they performed better than DCC8 and CDCC8, despite the larger number of boundary pixels to be coded.

The MPEG-4 Video VM 7 .0 framework for shape encoder, called Context based Arithmetic Encoder (CAE), has shown similar performance than the lossless technique presented in [7], which is based on DCC8. Thus, the OCU-LMGCC also intends for outperforming the MPEG-4 framework.

V. Concluding Remarks

This paper has provided significant contributions on the initial implementation of the algorithm MGCC [7], which culminated with the proposition of the OCU-LMGCC. The first kind of improvements have actually enabled the MGCC to be implemented, since the original algorithm presented apparent shortcomings. Other very important contributions introduced to optimize coding performance were: (i) the optimization of the parallel information required to eliminate ambiguity related to central pixel for lossless shape encoding; (ii) the presentation of novel switching schemes to determine the commutation between cell types and (iii) the proposition of an extended symbols set, better suited to the new switching schemes. These improvements all together provided gains at OCU-LMGCC of up to 32% in relation to the original lossless MGCC and 27.3% in relation to DCC8, whose performance is comparable to MPEG-4 framework according to [7].

Another important aspect to emphasize is that OCU-MGCC allows for faster decoding than DGCC, since it benefits from the fact that switching scheme to determine the cell type does not have to be run by the decoder.

Acknowledgment

The authors would like to thank to FIP/PUC Minas, PIBIC/CNPq and FAPEMIG for supporting this research and scholarships.

REFERENCES

[1] SHAO, B. et al; SMIL to Mpeg-4 BIFS Conversion. Proceedings Of The Second International Conference On Automated Production Of Cross Media Content For Multi-Channel Distribution, AXMEDIS ’06, pp 77-84, 2006.

[2] NUNES, P.; PEREIRA, F.; Object-Based Video Encoding. IEEE Transactions On Circuits And Systems For Video Technology. Volume 19 , Issue 9, pp 1274 - 1288, 2009.

[3] FUKUNAGA, I. et al; MPEG-4 Video Verification Model Version 12.1. ISSO/IEC JTC1/SC29/WG11, PEG98/N2552, Rome, December 1998.

[4] FREEMAN, H.; On the encoding of arbitrary geometric configurations. IRE TRANS. ELECTRON. COMPUT., Volume EC-10, pp 260-268, June, 1961.

[5] FREEMAN, H.; Computer processing of line drawing images. COMPUT. SURV. 6, pp 57-97, March, 1974.

[6] CHUNG, J.W. et al; Conditional Differential Chain Coding for Losless Representation of Object Contour. ELETRONICS LETTERS, Volume 34, Number 1, January, 1998.

[7] NUNES, P. et al; Multi-grid chain coding of binary shapes. IMAGE PROCESSING, 1997, PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON, pp 114-117.

[8] PARK et al; Compact Representation of Countors using directional grid chain code. SIGNAL PROCESS. IMAGE COMMUN., 2007, doi:10.1016/j.image.2007.11.002.