Scalable Ideal-Segmented Chain Coding

Mei-Chen Yeh, Yen-Lin Huang and Jia-Shung Wang

Department of Computer Science, National Tsing Hua University,
Hsinchu, Taiwan, R.O.C.
phone: 886-35-731067, fax: 886-35-723694
email: jswang@cs.nthu.edu.tw

ABSTRACT

In this paper, we present an optimal chain-code-like representation to code contour shapes; in addition, this representation can be easily extended to as a scalable form which structures shape data as the base layer followed by two enhancement layers. Lossy coding scheme is also presented for low-bit-rate applications. Compared with the block-based CAE method in MPEG-4 and DCC with arithmetic coder, our method not only has the higher compression ratio with less computation steps, but also can be applied to layered transmission.

The scalability achieved by our scheme can be recognized in both spatial and quality-wise. Compared with another scalable shape coding methods, our scheme is simple but more efficient than progressive polygon encoding methods. In case of the base layer with distortion $Dn = 0.02$, our scheme saves about 20~30% amount of bits, that is, we have a smaller size base layer, which is important in layered transmission.

1. INTRODUCTION

Binary shape coding, a new feature of MPEG-4 [1], is essential for efficiently coding and representing the arbitrary shape of any content object. In some applications, a user can access arbitrarily shaped objects in a scene and manipulate these objects interactively. In MPEG-4, block-based CAE method [6] is suggested to represent the contour shape. The CAE encodes the bitmap of shape using the context of the neighboring pixels by binary arithmetic encoding.

Chain code [2] is extensively adopted for representing contour information of an object. It basically requires two bits to record four directional links in a 4-connected contour shape. Several variations [3][4] were proposed to enhance the coding efficiency. Some involve the reduction of directional links, and the others include a variable length compression coder to further compress the code. Among them, Differential Chain Coding (DCC) is best known. DCC refers to a situation in which the differential between two consecutive pixels is small, and the trait is suitable for entropy coding. Conditional Differential Chain Coding (CDCC) [4] improves upon DCC in that the directional changes of successive contour pixels are related. Nevertheless, in CDCC and DCC, two bits are still required to represent a contour pixel before entropy coding.

In this paper, we present an optimal chain-code-like representation to code contour shapes; in addition, this representation can be easily extended to as a scalable form which structures shape data as the base layer followed by two enhancement layers. We introduce a specific kind of contours, which is referred to as the *ideal 4-connected chain*, in which only two directional links are required. And, an arbitrary contour shape can be decomposed into a sequence of such chains and related turning points, which is the junction between two consecutive chains.

The rest of the paper is organized as follows. Section 2 reviews conventional contour-based binary shape coding methods. Section 3 introduces our ideal-segmented chain code and our efficient algorithm. Section 4 presents a lossy method to further reduce the bit rate. The scalable version is discussed in Section 5. Section 6 presents and discusses the experimental results, which compare our methods with DCC and block-based CAE method adopted in MPEG-4. Conclusions are finally made in Section 7.

2. REVIEW OF CHAIN CODE AND DIFFERENTIAL CHAIN CODE

Chain code [2] is extensively adopted for representing contour information of an object. A 4-connected chain code consists of a starting point (x, y) and a sequence of directional links, $s_1, s_2, s_3, \ldots, s_n$, where $s_i = 0, 1, 2, 3$ and $n \geq 0$. The value "0" stands for RIGHT, "1" stands for UP, "2" stands for LEFT, and "3" stands for DOWN.

Original chain code requires 2 bits to record each directional link. To code effectively, the original chain code has several variations, some involving the reduction of directional links, the others including a variable length compression coder, such as Huffman code [5], arithmetic code [5], etc. In the former case, the simplest but famous one, referred to as differential chain code (DCC), records the move (forward, leftward and rightward) regarding two consecutive directional links instead of recording each link separately. It is not surprise that DCC with Huffman coding can reduce the required bit count comparing to that of the chain code. Generally, the average bit rate per directional link (contour pixel) is less than 1.5 for coding a smooth contour.

3. THE IDEAL-SEGMENTED CHAIN CODE (ISCC)

In this paper, we also address same problem to represent the chain code with less directional links. We first deal with the following question: Can we have a chain-code-like representation with the average bit rate per directional link equal to one? Definitely, the answer is unlikely to be true for arbitrary contour shape. However, a kind of specific contour shapes may perhaps satisfy this target, and an arbitrary contour shape comes

0-7803-7622-6/02/$17.00 ©2002 IEEE

out to be decomposed into a sequence of them. Accordingly, we define such specific kind of contours, which is referred to as the *ideal 4-connected chain*, as below.

Definition 1. A 4-connected chain $C = s_1, s_2, s_3, \ldots, s_n, n \geq 0$, is called an ideal 4-connected chain if both pairs of links (UP, DOWN) and (LEFT, RIGHT) do not appear in C simultaneously.

This definition reveals that an ideal 4-connected chain can move in the right-down (left-up) or left-down (right-up) fashion if tracing the contour in clockwise order. Figure 2 illustrates the typical contour profiles. For convenience, we denote these two cases as **Mode1** and **Mode 2**.

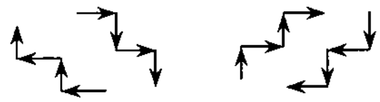

Figure 2. Typical contour profiles of Mode1 and Mode2.

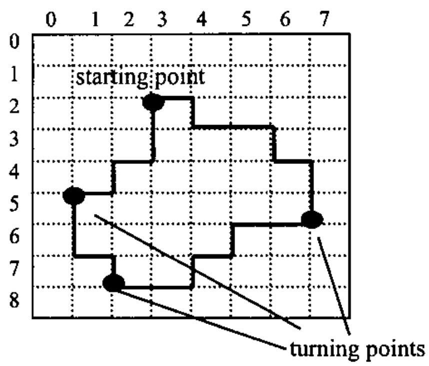

Figure 3. An illustrated example has four ideal chains.

As defined above, the direction links can be encoded by just one bit. Figure 3 illustrates a contour, which consists of four ideal chains.

Now, we discuss the decomposition problem of separating an arbitrary contour shape as a sequence of ideal 4-connected chains. We first define the term "*turning point*". Consider Fig. 3 again. Three turning points (TPs) and one starting point are located between two consecutive ideal chains. Once passing a TP, the mode will change. For example in Fig. 3, the mode transitions are Mode 1, Mode 2, Mode 1 and Mode 2, respectively. Formally, we have the following definition.

Definition 2. The turning point (TP) is between two consecutive ideal chains in a contour. In the case that the first ideal chain is Mode 1, the successive ideal chain is Mode 2, and the traverse order is clockwise, the turning point is positioned between the link "DOWN" and "LEFT", where "DOWN" is at the end of the first chain and "LEFT" is on the beginning of the second chain. The other cases can be defined similarly.

Notice that the number of turning points in an arbitrary contour, C, is fixed. That is, the number of ideal chains in the decomposition of an arbitrary contour is also fixed, which equals to the number of turning points plus one. We denote the number as $N(C)$. We conclude the above discussions and raise our new chain-code-like representation of the contour of an arbitrary object as below:

Definition 3. The *Ideal-Segmented Chain Code* (*IsCC*) of a contour C is an alternative sequence of points and ideal chains, $SP, I_1, TP_1, \ldots, TP_{n-1}, I_n$. where SP is the starting point, I_i is an ideal chain, TP_i is a turning point, and $n = N(C)$.

The total bit length of this representation is $\sum_{i=1}^{n} length(I_i)$ plus the code length of all TPs and SP. That is, the IsCC code could be optimal if the coding of all TPs is minimized.

Owing to coding each TP in terms of the two-dimensional coordinate is expensive, the location of each TP is recorded by the number of movements from the preceding TP. Consider the contour in Fig. 3 again, where the starting point is at (2,3), and the TPs are positioned at (6,7), (8,2) and (5,1), respectively. The one-dimensional contour indices are 8, 15 and 19; the corresponding differences are 8, 7, and 4. The optimal bit count for recording these three TPs is 8.

By the definition of ideal chain, it moves towards only two directions. Next, we give an efficient algorithm to decompose the contour to a sequence of ideal chains as well as code each of them all together. Note that our algorithm traces the contour in clockwise order.

Algorithm IsCC.

Input: A contour C of an arbitrary object.

Output: The Ideal-segmented chain code representation of C.

Step 1. Select the top point as the starting point. Initial mode is set as Mode1 and the default value of direction is RIGHT.

Step 2. Trace the contour in clockwise order. If the current direction is similar to the previous one, output a "0" and continue encoding the next move. Otherwise, go to step 3.

Step 3. If the mode condition does not change, output a "1". Otherwise, mark this point as a TP, output "0" and change the current mode.

Step 4. Trace the contour until go back to the starting point.

For example, the outcome sequence of the contour in Fig. 3 is **01101110001110011001110.**

Notice that some redundant TPs should be removed to further reduce the bit rate. Consider Fig. 4. TP **b** is superfluous because the chain from **a** to **c** is ideal. In this case, the chain **a** to **b** is denoted as Mode 1 instead of Mode 2. We denote such kind of TPs as *transient*. In the implementation, we need one extra bit to record this information.

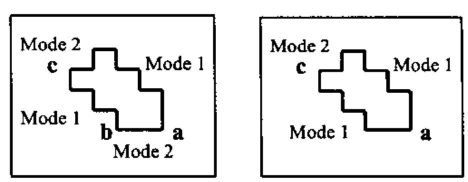

Figure 4. An example of redundant TPs.

4. LOSSY CODING

When a contour contains too many TPs, our algorithm might not perform well. The performance discussion is left in the section of experimental results. The reduction of the number of

TPs contributes the saving of the bit rate. However, the contour map changes due to this. We first check the contour shape between two consecutive TPs, and decide the possibility of removing some TPs without changing the shape drastically. In our experience, the case that two neighboring TPs are extremely close is happened often, possibly owing to the digitizing precision in the non-smooth contour maps. Figure 5 illustrates two typical examples.

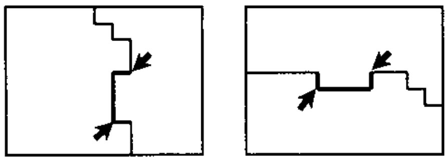

Figure 5. Two typical examples of adjoining TPs.

In case that the slope between two neighboring TPs is near vertical or horizontal, the digitized local shape can be viewed as vertical (or horizontal). Thus, two TPs can be dropped since the little local turning is minority. However, to ensure the fidelity, the threshold is in the range of subjectively tolerable to human eyes. Our lossy solution is easy, but having acceptable lossy contour shape compared with the block-based methods [6].

In order to classify the TPs corresponding to the lossy shape and the removed local shapes, we denote the former as the *global TPs* and the latter as the *local TPs*.

5. THE SCALABLE ISCC REPRESENTATION

In this section, a scalable version of IsCC coder is presented, which structures shape data as the base layer followed by two enhancement layers. This layered coding approach is useful in layered transmission. The scalable shape coding methods [6][7] were designed for this purpose. The coarse polygonal approximation is built in the base layer, each successive enhancement layer refines a given shape approximation by placing new vertices and perturbing existing vertices. The use of control points needs a large amount of computations. Our method is relatively simple and effective. Figure 8 illustrates the resulting outcomes of our scalable approach, coarse and fine shapes corresponding to the base layer and two enhancement layer approximations are depicted in (a), (b) and (c), respectively.

Recall that the global TPs, which correspond to the lossy encoding scheme, are regarded as key features of contour shapes. Our base layer consists of these global TPs and the *reduced ideal chains* between them. The so-called reduced ideal chain is a down-sampled version of the original ideal chain, which will be discussed later. The second enhancement layer, which is adopted to refine the coarse shape up to the lossless representation, consists of all local TPs and corresponding local ideal chains.

In layered coding, the base layer usually carries half or less amount of data. We shrink each ideal chain between two consecutive global TPs by simply a downsampling process. Several little turnings may be discarded owing to the shrinking. Figure 6 illustrates an example of shape changes in terms of the number of turnings. Partial shape of an ideal chain is shown in Fig. 6 (a), and Fig. 6 (b) demonstrates the downsampling process. The resulting (coarse) shape of the reduced chain is depicted in Fig. 6 (c). This coarse shape is dissimilar to the original one; it is suitable for plotting by lower resolution.

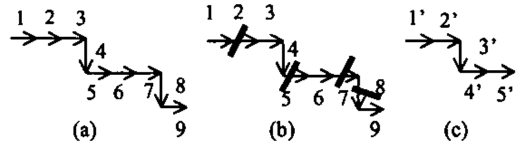

Figure 6. (a) original chain, (b) downsampling process, and (c) reduced chain in the base layer

The above downsampling process is intrinsically not convertible, thus more information is needed to help the recovery from reduced chain to the original one. This refinement information is stored in the first enhancement layer, which consists of the locations of eliminated turnings. Consider Fig. 6 again, step 4' in (c) should be marked since one turning is removed (step 8 in (b)). In the implementation, all marked steps are encoded by DPCM with modified Golomb coder [5], which is designed to encode integers with the assumption that the larger an integer the lower its probability of occurrence. Moreover, the first enhancement layer includes the odd/even information of each line segments that will be adopted in the upsampling process.

6. EXPERIMENTAL RESULTS

We conducted several experiments on demonstrating the performance of our IsCC as well as Scalable IsCC coders. We also justify them by comparing to the DCC with adaptive arithmetic coder and CAE method recommended in MPEG-4. Test sequences "Akiyo", "Bream2" and "Weather" are our test sequences, which are in QCIF format, and the resulting bit rates are calculated from the average bit count per VOP over 100 frames.

The performance of lossless shape coding is evaluated in terms of the number of encoded bits. Table 1 lists the average bits required for these methods. Bit rate reduction of IsCC over DCC is 8%-21%. The average bits per directional link of DCC can be reduced about 0.2 bits per contour pixel. Comparing to the CAE method in MPEG-4, our scheme significantly reduces the required bits, except "Weather". In the shape of "Weather", many TPs are located, accounting for the more bits than that of normal cases. Generally, the bits for contour-based shape coding methods are less than that of block-based methods. The proposed method encodes the outer contour pixels of the shape, whereas the CAE method encodes all of the pixels inside the shape.

Table 1. Average bit count / VOP – DCC with arithmetic, CAE and IsCC, 100 frames.

	DCC method	CAE in MPEG-4	IsCC
Akiyo	685.35	584.59	579.84
Bream2	880.96	808.18	696.65
Weather	568.53	494.41	523.53

A measurement is required to evaluate the distortion in both lossy and scalable encoding schemes. The quantity, *Dn*, which denotes the error pixel rate of an object, is adopted here to demonstrate the coding quality.

Table 2 lists the number of TPs and bit rate for three cases: with transient, without transient, and lossy. The shape of "Bream2" has 5 transient TPs since a lot of tiny zigzag curves it has. In shapes of "Akiyo" and "Weather", resulting lossy chains have comparative small number of global TPs because they have some long almost-vertical line segments that can be trimmed without losing the fidelity. Fig. 7 illustrates the reconstructed shapes and corresponding distortions.

Table 2. Number of TPs and bit rate for various coding schemes.

	Lossless, with transient TPs	Lossless, without transient TPs	lossy
Akiyo	11	8 (616)	4 (583)
Bream2	15	10 (644)	9 (633)
Weather	17	15 (595)	7 (518)

Figure 7. Lossy representation of 'Akiyo', $Dn = 0.002648$, 'Bream2', $Dn = 0.000295$, and 'Weather', $Dn = 0.003692$.

The effect of scalable coding is discussed here. Table 3 demonstrates the scalable IsCC encoding results of test sequences "Akiyo", "Bream2" and "Weather", frame thirteen. The base layer requires only half of the bits. In the shape of "Bream2", 3 layers totally takes much more bits because much of turning information are lost in the base layer, thus the first enhancement layer requires more bits to recover them. In the shape of "weather", the required bits for encoding all three layers are even less than that of the lossless representation. The reason is that the shape is relatively smooth and the upsampling process gains the high-quality prediction in recovering the contour.

In Fig. 8, resulting shapes corresponding to the base (1/3), the first (2/3) and second (3/3) enhancement layers are presented. They require 308, 520 and 584 bits, respectively, and the corresponding bit errors are: 134, 26 and 0.

Table 3. Scalable coding results of Akiyo, Bream2 and Weather.

		Akiyo	Bream2	Weather
1/3 layer	bits/VOP	311	386	308
	Dn	0.017055	0.021983	0.020615
2/3 layers	bits/VOP	567	888	520
	Dn	0.001695	0.000295	0.004
3/3 layers	bits/VOP	620	901	584
	Dn	0	0	0
Lossless	bits/VOP	616	644	595

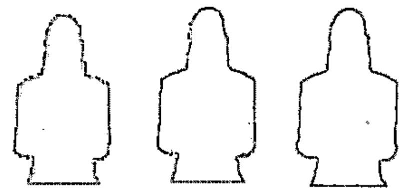

Figure 8. Shapes correspond to (a) the base layer (b) the first enhancement layer and (c) the second enhancement layer. Black pixels represent the errors.

The scalability achieved by our scheme can be viewed in both spatial and quality-wise. While decoding the base layer only the reduced shape is a sub-sampled version of the original one (spatial scalability). It can be upsampled to the original size, leading to an approximated reconstruction (quality scalability). Compared to another scalable shape coding methods, entitled the *progressive polygon encoding methods* [6][7], our scheme adopts turning points and the ideal chains to support scalability, which is simple and more efficient than progressive polygon encoding methods which successively transmit the polygonal approximation refinements. To achieve the quality for the base layer with 0.02 *Dn* value, our scheme saves about 20~30% amount of bits.

7. CONCLUSIONS

A simple but efficient method for coding the contour shape of an arbitrary object was presented in this paper. It is based on the chain code representation that utilizes the supports of ideal chain and the decomposition of a contour shape as a sequence of turning points and ideal chains. This representation is optimal. Several kinds of TPs are investigated, such as global, local and transient, and they can contribute to the scheme design of lossy and scalable. To provide the layered transmission of object shape through channels of various bandwidths, the scalable version was discussed in this paper also.

REFERENCES

[1] MPEG-4 Video Verification Model Version 13.0, ISO/IEC JTC1/SC29/WG11 N2687, Mar. 1999.

[2] H. Freeman, "On the ending of arbitrary geometric configurations," *IRE Trans. on Electronic Computers*, Vol. EC-10, pp. 260-268, June 1961.

[3] T. Kaneko and M. Okudaria, "Encoding of arbitrary curves based on the chain code representation," *IEEE Trans. on Communications*, Vol. 33, pp. 697-707, July 1985.

[4] Jae-Won Chung, Joo-Hee Moon and Jae-Kyoon Kim, "Conditional differential chain coding for lossless representation of object contour," *Electronics Letters*, Vol. 34, pp. 55-56, Jan. 1998.

[5] Khalid Sayood, *Introduction to Data Compression*, Academic Press, Second Edition, 2000.

[6] A. K. Katesagglos, L. P. Kondi, F. W. Meier, J. Ostermann, G M. Schuster, "MPEG-4 and ratio distortion based shape coding techniques," *Proc. IEEE*, Vol. 86, pp.1126-1154, June 1998.

[7] Gerry Melnikov, Aggelos K. Katesaggelos, "A rate-distortion optimal scalable vertex based shape coding algorithm," *Proc. ICASSP*, Vol. 4, pp.1947-1950, 2000.