Motion Compensated Two-link Chain Coding for Binary Shape Sequence

Zhitao Lu
William A. Pearlman
Electrical Computer and System Engineering Department
Rensselare Polytechnic Institute Troy NY 12180
luz2, pearlw@rpi.edu

ABSTRACT

In this paper, we present a motion compensated two-link chain coding technique to effectively encode 2-D binary shape sequences for object-based video coding. This technique consists of a contour motion estimation and compensation algorithm and a two-link chain coding algorithm. The object contour is defined on a 6-connected *contour lattice* for a smoother contour representation. The contour in the current frame is first predicted by global motion and local motion based on the decoded contour in the previous frame; then, it is segmented into motion success segments, which can be predicted by the global motion or the local motion, and motion failure segments, which can not be predicted by the global and local motion. For each motion failure segment, a two-link chain code, which uses one chain code to represent two consecutive contour links, followed by an arithmetic coder is proposed for efficient coding. Each motion success segment can be represented by the motion vector and its length. For contour motion estimation and compensation, besides the translational motion model, an affine global motion model is proposed and investigated for complex global motion. We test the performance of the proposed technique by several MPEG-4 shape test sequences. The experimental results show that our proposed scheme is better than the CAE technique which is applied in the MPEG-4 verification model.[1]
Keywords: Shape Coding, Video Coding, Chain Coding, MPEG-4, Motion Estimation, Motion Compensation.

1. INTRODUCTION

With the emergence of multimedia applications, functions such as access, searching, indexing and manipulation of visual information at the semantic object level, are becoming very important issues in research and standardization efforts, such as MPEG-4. In MPEG-4, each object is represented by three sets of parameters, shape, texture, and motion so that the object can be encoded, accessed and manipulated in arbitrary shape. Among these three sets of parameters, shape information is crucial for object representation and object-based coding. In order to transmit the shape of an object efficiently, a large number of techniques have been proposed.[1–7]

The shape coding techniques are typically classified as block-based techniques and contour-based techniques. Block-based techniques use a binary image to represent the shape of the video object; this binary image is encoded block by block as in the conventional image coding technique. The context-based arithmetic encoding (CAE)[1] is one of the most successful methods for binary image coding and has been adopted in the MPEG-4 verification model. Contour-based techniques perform the compression along the boundary of the video object. A polygon or a contour list is usually used to represent the shape of a video object. For these representations, distortion between the decoded and original shape information is easy and well defined. According to the specified distortion, the coding algorithm can achieve lossy and/or lossless coding.

The shape images bear a high degree of similarity along the temporal dimension of an image sequence. The coding efficiency can be improved by exploiting the temporal redundancy in a shape sequence.[3,5]

In this paper, we present a motion compensated two-link chain coding technique. The contour of the video object is defined on a 6-connected contour lattice for a smoother contour representation. A tow-link chain coding technique is proposed to exploit the spatial redundancy within the object contour. In contour motion estimation and compensation, besides translational global motion model, we also investigate an affine global motion model for complex motion in the contour sequence. The whole paper is organized as follows: In section 2, we present the two-link chain coding technique. In section 3, both translational and affine global contour motion estimation are presented. The experimental results on MPEG-4 shape sequences are presented in section 4. The paper is concluded in section 5.

2. TWO-LINK CHAIN CODE

Our approach is a modified chain coding method which belongs to the contour-based technique. We define the contour points on the *contour lattice* which consists of the half-pixel positions on the image. As illustrated in Fig. 1, the ∘ represents the original image pixel; the $*$ and $+$ represent the half-pixel positions between two neighboring image pixels in the horizontal and vertical directions; The box represents the remaining half-pixel positions. Throughout the whole paper, *top* position, *left* position and *vertex* position are used to refer to these three types of half-pixel positions.

In our chain coding system, A contour point is on a *top* (or *left*) position when its two neighboring image pixels in the vertical (or horizontal) direction have different labels. Any contour which passes a *vertex* position can be represented by a contour without passing that *vertex* position. Therefore, our contour lattice only contains the *top* and *left* half-pixel positions. One advantage of this contour lattice representation is that it makes contour smoother.

In order to improve the contour coding efficiency, we impose constrains on the contour to be encoded by applying a majority filter.[3] A majority filter simplifies the contour by removing its rugged components. The effect is similar to the perfect 8-connectivity constraint.

Another feature of our chain code is the joint encoding of two consecutive links in oder to exploit the spatial redundancy of the contour. An illustration of the encoding of two consecutive links is shown in Fig. 1. On the *contour lattice*, each contour point has 6 possible links to its six neighbors. After majority filtering, the number of the next possible links is more limited. If the current *link* is in the horizontal or vertical direction, there are 3 possible directions for the next *link*. If the current *link* is in a diagonal direction, there are only 2 possible directions for the next *link*; other directional links are not possible. If the last link is in the horizontal or vertical direction, there are 7 possible combinations for the next two contour links; If the last link is in the diagonal direction, there are 5 possible combinations for the next two contour links. In order to reduce the bit-rate for the straight contour segment, we add two more dashed links. When the contour segment is a straight line in a diagonal direction, one code can represent 3 or 4 contour links. Without using an entropy encoder, the bit-rate of the proposed chain code is ≤ 1.5 bits/link.

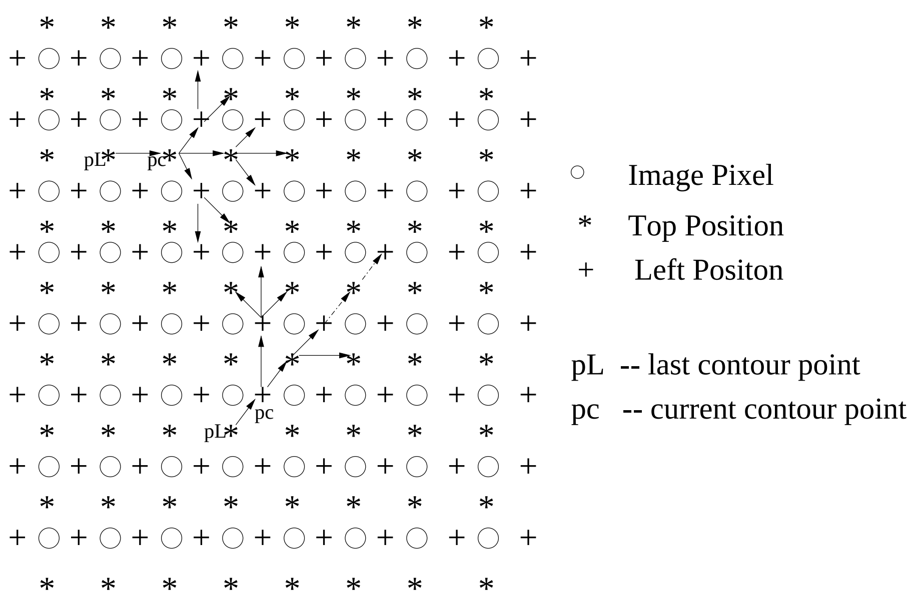

Figure 1. Proposed two-pixel chain code

We use a context-based arithmetic encoder to encode the chain code sequence for a higher coding efficiency. The context is the direction of the last contour link. In total, there are 12 contexts, six of them begin from a *top* position, and the other six begin from a *left* position. The probability of each codeword under each context is adapted during the coding procedure.

3. CONTOUR MOTION ESTIMATION AND COMPENSATION

Since a contour sequence has very high correlation in temporal domain as the texture does, a straightforward method to exploit its temporal redundancy is using motion estimation and compensation. The contour in the current frame can be predicted from the contour obtained in the previous frame. Only the contour segment which can not be predicted is encoded by the shape coding technique. This can reduce the bit-rate of shape coding drastically. In the literature, the contour motion is usually assumed as translational motion. This approach works well for image sequences with low speed or simple motion. When there is zooming and/or rotation, this assumption does not work well. In this paper, we use two global motion models, translational global motion model and affine global motion model, to predict the global motion of the contour sequence to be encoded.

3.1. Translational Contour Motion Estimation

In this contour motion estimation/compensation scheme, the object contour is assumed to undergo a translational motion. A global motion vector for the whole object contour is searched according to the number of matched contour points between two contours. The whole contour is segmented into *global motion success* segments and *global motion failure* segments. For each *global motion failure* segment, a *local motion vector* search process is applied. This process searches for the motion vector that which minimizes the following function:

$$f(mv) = bit(mv) + bit(length) + bit(failuresegment) \tag{1}$$

where $bit(mv)$ denotes the number of bits used for the local motion vector mv; $bit(length)$ the number of bits used for the length of the local motion success segment under motion vector mv; and $bit(failuresegment)$ the number of estimated bits used for the motion failure segment under motion vector mv.

By using this function, the small motion success segment which consume more bits to be encoded can be avoided.

3.2. Affine Contour Motion Estimation

When there is more complex motion such as *zoom* and/or *rotation*, the contour can not be well compensated by translational motion model. Here we investigate an affine global motion model for these cases. We use the following six-parameter affine motion model as the global contour motion model.

$$\hat{x} = a_1 x + a_2 y + a_3 \tag{2}$$

$$\hat{y} = a_4 x + a_5 y + a_6 \tag{3}$$

In the above equations, $\hat{x}$ and $\hat{y}$ are the coordinates of contour points in the current frame. x, y are the coordinates of contour points in the previous frame.

The problem is to estimate the vector $[a_1, a_2, a_3, a_4, a_5,\ a_6]$ according to available contours. As shown in Fig. 2. First the corner points of each contour are detected according to their curvature values. Then, the corner points are matched by a corner matching process.[8] The motion vectors are calculated from these matched corner pairs. The affine parameters are estimated by a least median square algorithm.[9]

In order to test the corner matching and the affine motion estimation algorithms, we create shape images undergo translational and rotational motion from an available shape image. The actual motion parameters and the estimated motion parameters are listed in Table 1. The number of correctly predicted contour points by the decompressed motion parameters are listed in Table 2. We can see that the estimated affine parameters are fairly accurate.

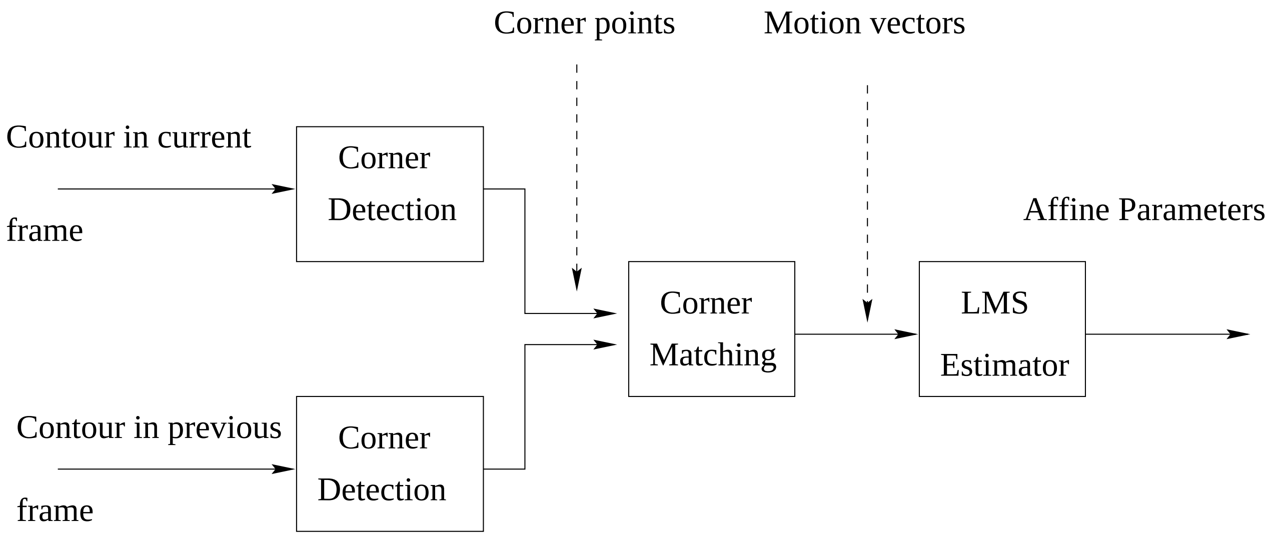

Figure 2. Diagram of the affine motion estimation for a contour sequence

Table 1. True and estimated affine parameters

	a_1	a_2	a_3	a_4	a_5	a_6
True	0.	-.055	-5.	.055	0.	5
Est	.0017	-.054	-5.07	.059	-.0007	3.9
True	0	0	-5	0	0	5
Est	0.	0.	-5.	0	0	5

Table 2. Number of pixels which are correctly predicted

	Correct Pixels	Total Pixels
Translational	129	514
Affine	408	514

3.3. Proposed Motion Compensated Chain Coding System

Two coding modes: the *Intra* mode and the *Inter* mode are used in our shape coding scheme. The contour in the first frame within a GOP (group of picture) is encoded by the *intra* mode. The contours in other frames are encoded by the *inter* mode. The diagram of the *inter* mode contour coding is shown in Fig. 3. In the *intra* mode coding, the contour is encoded directly by the two-link chain coding method. In the *inter* mode coding, the contour is first predicted by the motion estimation and compensation, then only the motion failure segments are encoded. The syntax of the bitstream is shown in Fig. 4. The first bit is the global motion flag, followed by the global motion vector, the coordinate of the start point, and then the bitstream for each contour segment. For a global motion success segment, only its length is encoded. For a local motion success segment, both its length and the motion vector are encoded. For a motion failure segment, a series of chain codes are followed. An *END* is used to tell the decoder that the end of this segment is reached.

4. EXPERIMENTAL RESULTS

We test the performance of the proposed algorithm by coding several widely used MPEG-4 test shape sequences: the Akiyo and Weather sequences in QCIF and CIF format with frame rate of 30 fps. The total length of the sequences is 300 frames. We code the sequences in both the *intra* mode and the *inter* mode. First the shape

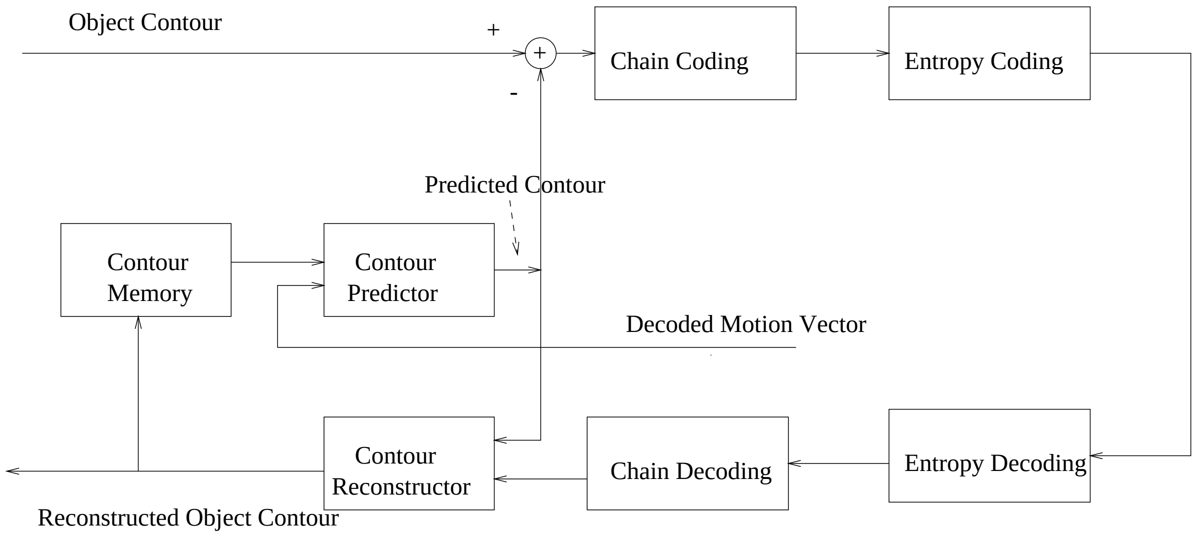

Figure 3. Diagram for the inter mode contour coding

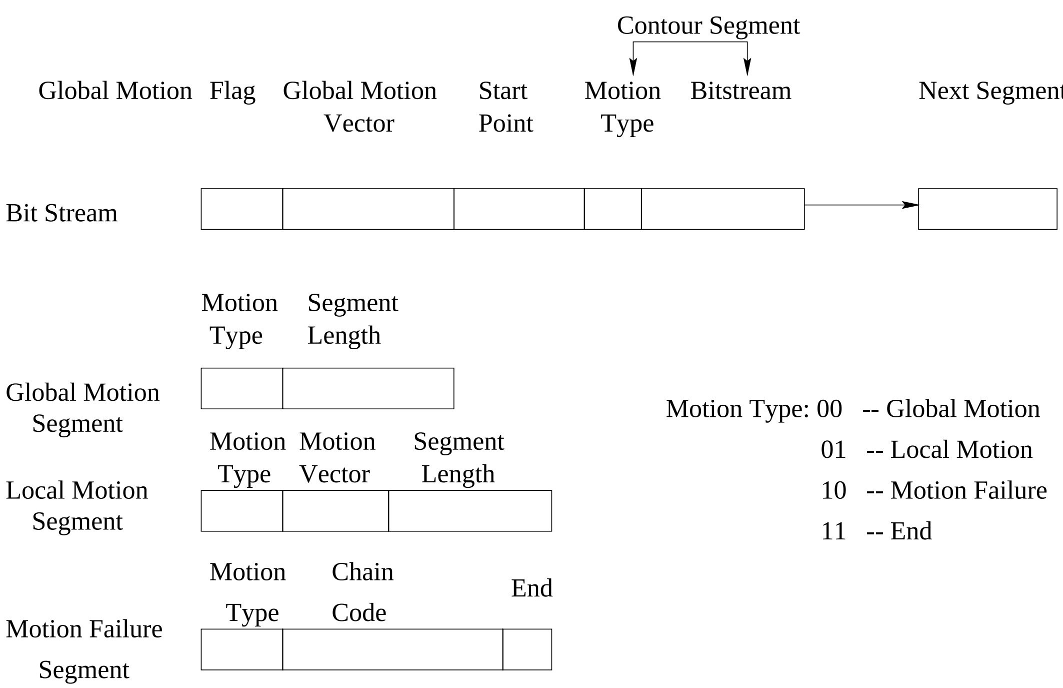

Figure 4. Bitstream syntax for contour coding

image is transformed into object contour defined on the *contour lattice*. Then the contour links are encoded by the proposed chain coding technique.

We compare the performance of our proposed shape coding technique with the CAE and baseline-based coder.[7] The results are listed in Table 3. From Table 3, the performance of our algorithm is better than that of the CAE and the Baseline-based technique.

Table 3. Comparison of shape coding techniques in intra mode

Sequence	Frame	Format	CAE	Baseline	Proposed algorithm
Akiyo	0	QCIF	0.06059	N/A	0.0576
Akiyo	0	CIF	0.03657	N/A	0.0260
Weather	0	QCIF	0.05581	N/A	0.0574
Weather	30	QCIF	0.0801	0.0745	0.0579
Weather	0	CIF	0.03167	N/A	0.0252

We also compare the performance of our proposed shape coding technique with the CAE and another motion compensated contour-based technique, GPSC.[5] The results are listed in Table 4. From Table 4, the performance of our algorithm is better than that of the CAE and GPSC technique.

Table 4. Comparison of shape coding techniques in inter mode (bits/frame)

Sequence	Format	frame rate	CAE	GPSC	Proposed algorithm
Weather	QCIF	30fps	303	N/A	288
Weather	QCIF	10fps	382	394	356

5. CONCLUSION

In this paper, we present our new motion compensated two-link chain coding method for 2-D shape coding. The contour points are defined on the *contour lattice*, a 6-connected image. We impose a smooth constraint on the object contour by applying a majority filter to improve the coding efficiency. In the inter mode, we investigate the translational and affine-model based motion compensated chain coding scheme for shape sequence coding. The experimental results show that affine motion model works well for more complicated global motion than traditional translational motion model. The experimental results show that our proposed scheme uses fewer bits than that of the CAE technique which is applied in MPEG-4 VM7.0 and other contour-based techniques: GPSC and Baseline-based technique.

REFERENCES

1. N. Brady, F. Bossen, and N. Murphy, "Context-based arithmetic encoding of 2d shape sequences," in *Special session on shape coding, ICIP97*, 1997.
2. H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, vol. 10, pp. 260–268, June 1961.
3. C. Gu and M. Kunt, "Contour simplification and motion compensated coding," *Signal Processing: Image Communications (Special Issue on Coding Techniques for Very Low Bit-Rate Video*, vol. 7, pp. 279–296, Nov. 1995.
4. T. Kaneko and M. Okudaira, "Encoding of arbitrary curves based on the chain code representation," *IEEE Trans. on Communications*, vol. COM-33, pp. 697–706, July 1985.

5. J. I. Kim, A. C. Bovik, and B. L. Evans, "Generalized predictive binary shape coding using polygon approximation," *Signal Processing: Image Communication*, pp. 643–663, July 2000.
6. L. Labelle, D. Lauzon, J. Konrad, and E. Dubois, "Arithmetic coding of a lossless contour-based representation of label images," in *Proc. of the International Conf. on Image Processing*, vol. 1, 1998.
7. S. H. Lee, D.-S. Cho, Y.-S. Cho, S. H. Son, E. S. Jang, J.-S. Shin, and Y. S. Seo, "Binary shape coding using baseline-based method," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, pp. 44–58, Feb. 1999.
8. R. N. Strickland and Z. Mao, "Computing correspondences in a sequence of non-rigid shapes," *Pattern Recognition*, vol. 25, pp. 901–912, Sept. 1992.
9. P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection.* New York, NY: John Wiley Sons, 1987.