A contour coding algorithm using DST

Jong Lak Kim, Jin Hun Kim, Choong Soo Park, and Han Soo Kim

Video Research Center, DAEWOO Electronics CO., LTD.

Abstract

Abstract: In this paper, an efficient contour coding algorithm incorporating polygonal approximation and discrete sine transform is introduced. Contour information is inevitable in content based coding, and polygonal approximation method is widely used to compress the contour information. However polygonal approximation method is not suitable when fine contour is needed. We show that the error signal of polygonal approximation can be efficiently represented using DST, that is, the contour information can be represented accurately with polygons and DST coefficients. With this contour coding scheme, the required bits to represent a contour can be reduced by about 40-50% with virtually no degradation compared to the existing chain coding method.

I. Introduction

Some emerging applications involving interactive multimedia services need the ability to efficiently access and represent independent visual objects in a scene. In general, this requires the ability to represent arbitrarily shaped objects. Key components of the object representation are the coding of the shape, the region texture, and the motion. This paper addresses the first of these three components: the contour coding method.

Chain coding method [1,2] provides a lossless compression of the shape information, but requires too many bits to be used in very low bit rate coding scheme. Other contour coding methods, such as polygonal approximation method [1,2], require fewer bits. However, these lossy compression schemes may introduce visible distortions.

In the proposed contour coding method, object contours are represented by a combination of approximated polygons and contour errors, where contour error represents the difference between the original contour and the approximated polygon. To represent contour error, 1-dimensional DST (discrete sine transform) [1] is adopted. Several simulation results reveal that the proposed contour coding method could achieve significant improvement in the coding efficiency over lossless chain-coding method and lossy polygonal approximation method.

II. The proposed contour coding method.

The proposed contour coding scheme is composed of five steps: preprocessing, polygonal approximation, error compensation using DST, quantization, and vertex coding. Preprocessing is required before polygonal approximation to refine the contour image. In this process, the false contour is eliminated. The false contour signifies short terminal branches and short contour segments which are not connected to any other contours. Figure 1 shows the typical examples of false contours.

After preprocessing, contour is approximated by polygons and the approximation error signal is coded using DST. In addition, the vertices of polygons are coded using both fixed length code and syntax based arithmetic code.

Now, we will describe some major procedures more thoroughly.

II.1 Polygonal Approximation

The procedure for polygonal approximation is as follows:

1. Determine the threshold for maximum distance between the line segment and the original contour segment.
2. If the contour is not a closed loop, then initially approximate the contour with a line segment which connects the two end points, A', B', as shown in Figure 2(a). If the contour is a closed loop, then set the farthest two points A', B' as vertices, as shown in Figure 3(a).
3. Find the farthest point on the contour which has maximum distance from corresponding line segment.
4. If the maximum distance is larger than the given threshold, divide the corresponding contour segment into two new segments at the

farthest point as shown in Figure 2(b), 2(c), 3(b), 3(c).

5. Approximate the new contour segments as line segments.
6. Repeat step 3, 4, and 5, until all distance between the line segment and the pixels of the corresponding contour segment does not exceed the threshold.

II.2 Error compensation using DST

The polygonal approximation error signal is sampled at eight positions, regardless of the length of the line, as shown in Figure 4. Then, the eight error samples are transformed using DST. The pair of DST kernel is as follows:

$$E(k)=\sqrt{\frac{2}{N+1}}\sum_{n=0}^{N-1}e(n)\sin\frac{\pi(k+1)(n+1)}{N+1}$$

$$e(n)=\sqrt{\frac{2}{N+1}}\sum_{n=0}^{N-1}E(k)\sin\frac{\pi(k+1)(n+1)}{N+1} \qquad (1)$$

where N is the number of samples (here, N=8), and $e(n)$ is the sampled error signal shown in Figure 4 and $E(k)$ is the DST coefficients. To encode the DST coefficients, we quantize the coefficients with a uniform quantizer. Let D_{max} be the given maximum distance, then, generally, the maximum value of $E(k)$'s does not exceed $2 \cdot D_{max}$. Thus, if we denote the number of quantization steps as M (where M is an odd number), then the step size of the quantizer can be given as $4 \cdot D_{max}/(M+1)$.

II.3 Vertex coding

In order to encode vertex information, the relative positions of vertices are coded using both Fixed Length Code (FLC) and Syntax-based Arithmetic Code (SAC) [3,4]. Note that the relative spacing of the object's vertices are considered in order to produce a compact representation of the vertices. The method consists of adapting the representation at an object level by determining the dynamic range of the relative locations of the object's vertices, further adapting the representation for each vertex by exploiting the use of SAC algorithm.

The procedure for vertex coding is as follows:

a. Relative-Location Dynamic Range Coding

Here, $\overline{V}=\{V_0,V_1,\dots V_{N-1}\}$ represents the ordered set of N vertex locations approximating an object, and $\overline{R}=\{R_i=V_i-V_{i-1}\ for\ i=1,\dots N-1\}$ represents the relative locations of the vertices. The dynamic range of the relative locations of the object's vertices are determined and indicated in the compressed bitstream by:

1. calculating the relative locations of the vertices, $R_i=V_i-V_{i-1}\ for\ i=1,\dots N-1$,
2. determining *x_max_magnitude* and *y_max_magnitude*, the maximum absolute values of the x and y components of the R_i, respectively,
3. determining the group information for *x_max_magnitude* and *y_max_magnitude*, which are needed for SAC,
4. encoding the determined group information with residual upper bits excluding lower 3 bits using Fixed Length Code (FLC). Because the upper bits have some trends, if we use entropy coding technique for them, we can improve the coding efficiency.

For example, let us consider an example in which the set of vertices approximating an object consisted of $\overline{V}$= (80,60), (105,66), (85,70), and (70,63). In that case, $\overline{R}$ = (25,6), (-20,4), and (-15,-7), *x_max_magnitude* = 25, *y_max_magnitude* = 7, and suppose CIF image, then *x_max_magnitude* can be represented by 000011001. Now we only transmit number of group which add one to upper 6 bits excluding lower 3 bits, i.e., 000100. In this example we have four groups, 0,1,2, and 3. We can also encode the number of group for *y_max_magnitude* with same way.

b. Locally-Adaptive SAC-Based Vertex Representation

Each vertex is then encoded taking into account the dynamic ranges of the x and y components of the relative locations of each vertex, except for the first vertex, V_0, whose absolute address is coded. The representation is further adapted at each vertex by exploiting syntax-based arithmetic coding (SAC) algorithm. The following steps are used to encode $R_i=V_i-V_{i-1}\ for\ i=1,\dots N-1$, i.e., the relative location of each vertex after the initial vertex:

1. Determine the group of R_i. The group numbers are defined as residual upper value excluding lower three bits.
2. Encode the group number using Variable Length Code (VLC).
3. Encode the excluded lower three bits are coded with Fixed Length Code (FLC).
4. Encode the indicator for each vertex which represents either end vertex of contour or not.

3. Simulation Results

Here, we compared two contour coding methods: the proposed contour coding method and the existing polygonal approximation method [1]. Note that the proposed vertex coding method is applied in both cases. The input contour is shown in Figure 5(a), 6(a).

In the proposed contour coding method, the number of sample points for each segment is eight, and the number of steps for quantizer is seven. If the number L of pixels on a contour segment is smaller than 9, the higher (9-L) coefficients are set to zero to prevent the redundant coefficients from being transmitted. D_{max} are quantized with 10 bits.

To compare the results, we defined the error pixels as the pixels in the area of the gap between the false contour and the correct contour. As shown in Figure 5, 6, the subjective gain of the proposed contour coding scheme over polygonal approximation method is noticeable. Note that the same amount of bits are used in both Fig. 5(a) and Fig. 5(b), also, in Fig. 6(a) and Fig. 6(b). In Fig. 7 and Fig. 8, we presented the variance of error pixels with respect to the number of bits per contour pixel. It is clear that the more we reduce the number of bits per contour pixel, the more efficient the proposed contour coding method becomes. This stems from the fact that the DST coefficients compensates the polygonal approximation errors quite well with only a small amount of additional information.

4. Conclusion

The proposed contour coding scheme exploiting polygonal approximation, DST, and SAC is very efficient in representing arbitrarily shaped boundary information. This mainly stems from the fact that the DST kernels resemble the polygonal approximation error signal in most cases. As shown in Fig. 5(c), 6(c), the errors are not visible when the contours are encoded with about 40% of the bits required by the chain coding employing JPEG binary arithmetic code [5]. Note that, in the proposed contour coding method, about 70~80% of the required bits are consumed to represent the vertex information.

References

[1] Anil K. Jain, *Fundamentals of digital image processing*, Prentice-Hall International, Englewood Cliffs, NJ, U.S.A., 1989.

[2] Theo Pavlidis, *Algorithms for graphics and image processing*, Computer science press, Rockville, Md, U.S.A., 1982.

[3] Ian H. Witten, Radford M. Neal, and John G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, No. 6, June 1987.

[4] ITU-T, *Draft H.263, Video coding for low bitrate communication*, Dec. 1995.

[5] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*, Van nostrand reinhold, NY, U.S.A., 1993.

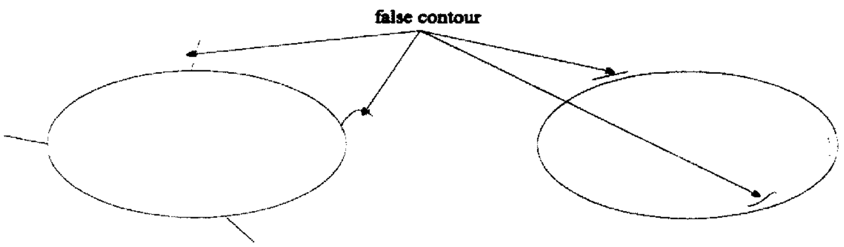


Figure 1. Examples of false contour.

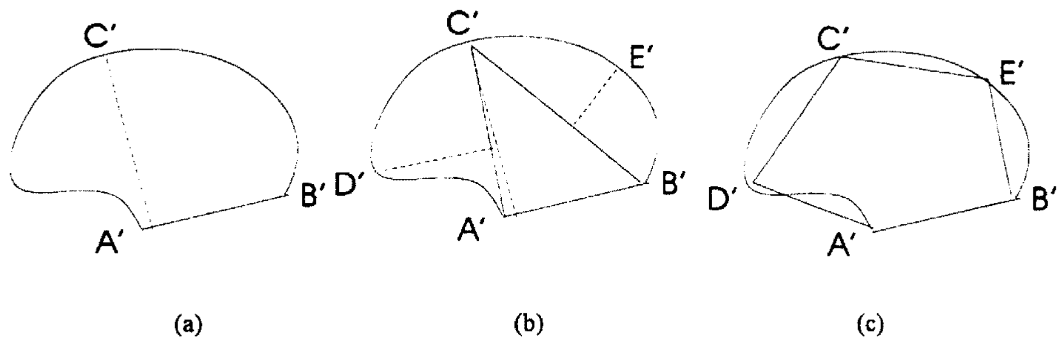


Figure 2. Segmentation of an open contour.

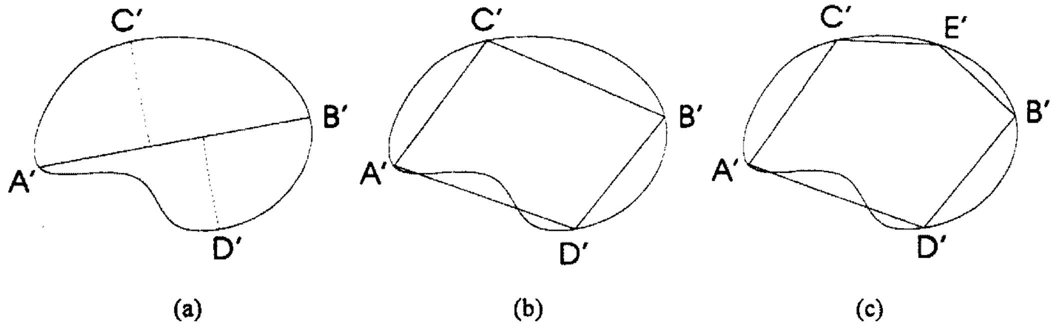


Figure 3. Segmentation of a closed contour.

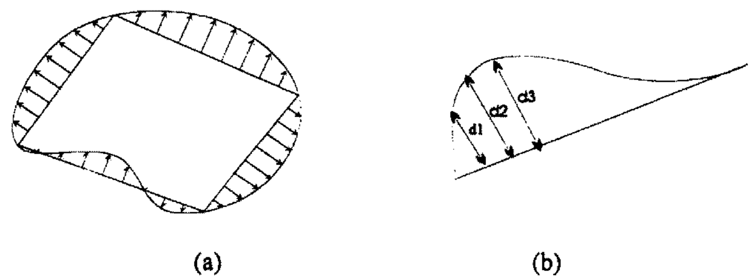
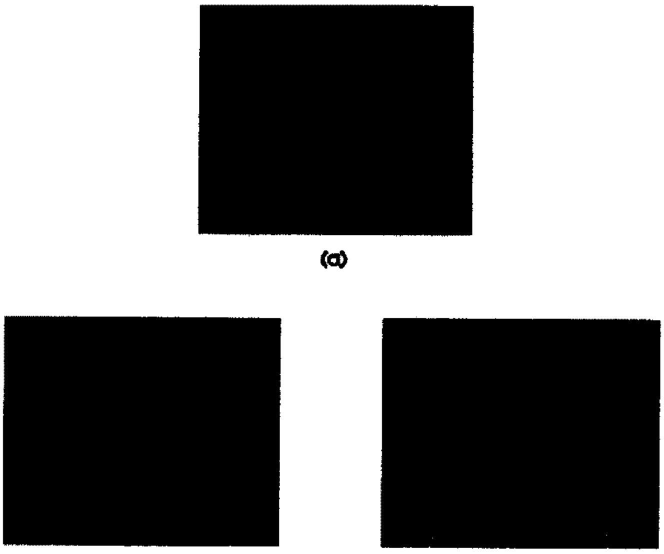


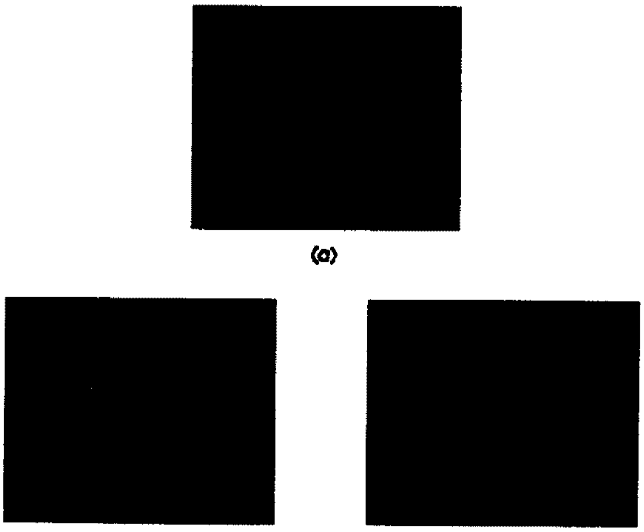
Figure 4. Example of a sampled error signal (for Figure 4(b))



(a)

(b) (c)

Figure 5. Contour images of CLAIRE,
(a) original (b) polygonal approximation method (c) proposed method.



(a)

(b) (c)

Figure 6. Contour images of MISS AMERICA,
(a) original (b) polygonal approximation method (c) proposed method.

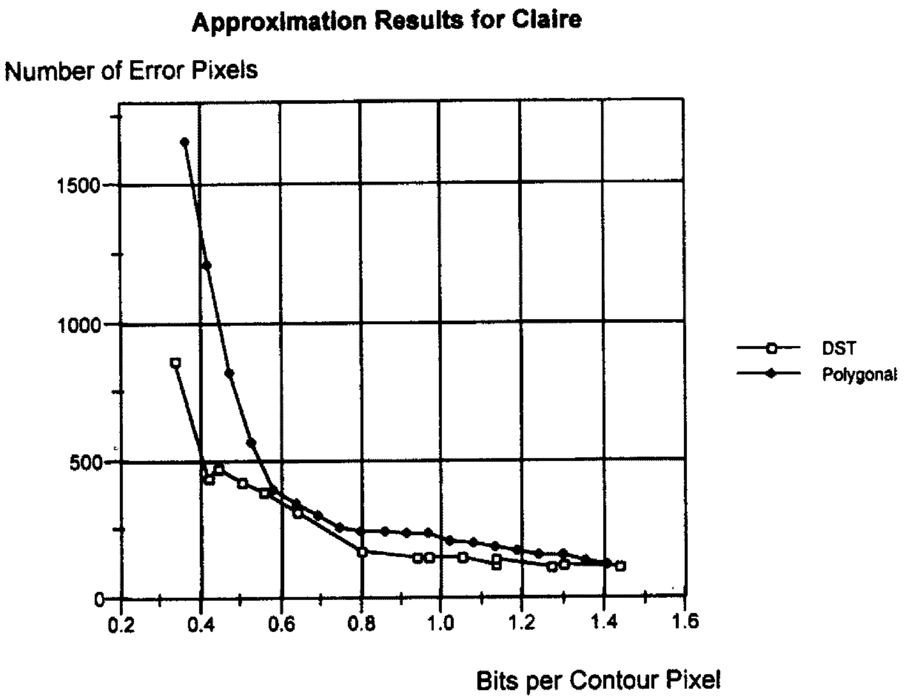


Figure 7. Comparison of contour coding methods on CLAIR

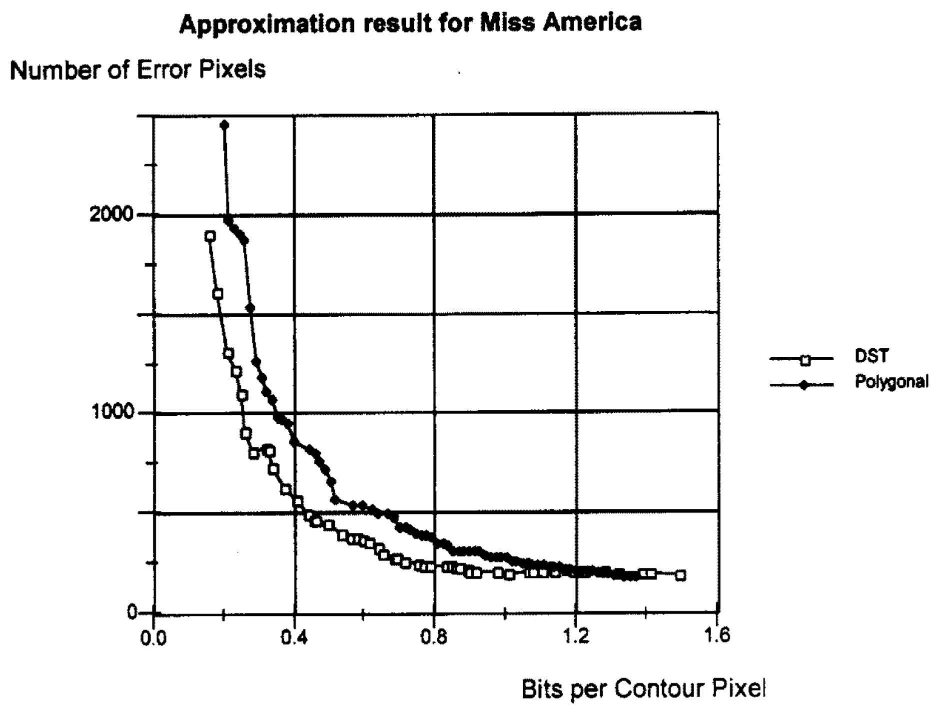


Figure 8. Comparison of contour coding methods on MISS AMERICA.