Boundary Block-Searching Algorithm for Arbitrary Shaped Coding

Chia H. Yeh[1] and Hsuan T. Chang[2], Member SPIE

[1]MAVs Lab. Inc.
Lung-Tan Tau-Yuan, 325 Taiwan ROC

[2]Photonics and Information Laboratory
Department of Electrical Engineering
National Yunlin University of Science and Technology
Douliu Yunlin, 64045 Taiwan ROC

Abstract

Arbitrary shaped coding is an important issue of MPEG-4. In this study, an efficient shaped coding method, called the boundary block-searching (BBS) algorithm that can enhance the coding efficiency of conventional padding schemes, is proposed. The proposed BBS algorithm assumes that boundary blocks have strong correlation even though they are not connected. For an input boundary block, the most similar block (only object pixels are considered) is sought from the previously coded data. Instead of being encoded by the use of discrete cosine transform, here the boundary block is encoded by a position vector, which indicates the relative position of the most similar block. Therefore, the number of bits required to denote the boundary block is greatly reduced and low bit rate can be achieved. For two video sequences under different test conditions, simulation results show that the proposed BBS algorithm can greatly improve the coding efficiency.

Subject terms: Arbitrary shaped coding, object-based coding, MPEG-4, boundary blocks, and LPE.

1 Introduction

The main difference of the MPEG-4 [1] standard compared with H.261 [2], MPEG-1 [3], MPEG-2 [4] and H.263 [5] is that MPEG-4 supports object-based video coding in addition to block-based coding techniques. MPEG-4 allows the transmission of arbitrary shaped video objects (VOs) [6]–[9]. In general, VOs have arbitrary shapes and contain both background and object pixels. Thus, the efficient coding method for objects pixels is the prerequisite condition to achieve MPEG-4 [10]–[15].

Boundary blocks of VOs consist of both the background and object pixels, where the shape information is provided. For boundary blocks of VOs, the background region is filled with some appropriate values before applying the discrete cosine transform (DCT) operation for coding. The repetitive and the low pass extrapolation (LPE) padding schemes [16], [17] are two famous methods for the boundary block coding in MPEG-4. For the repetitive padding scheme, the background region is filled with extending boundary pixel values, while for the LPE padding algorithm, the background region is filled with the average of object pixel values, and then a low pass filtering is applied only to the background region. Because these padding-based methods are not the optimal solution, further refined methods, such as the shape adaptive DCT [18], [19], for the arbitrary shaped coding were proposed. The simplest method among various kinds of padding-based coding schemes is the repetitive padding scheme, which extends boundary pixel values to fill the background region [16]. The repetitive padding scheme extends the boundary pixels via three methods: the horizontal, vertical, and averaging padding methods. The merit of the repetitive padding scheme is the lowest computational complexity but the coding performance is very low.

Another padding method is the LPE padding algorithm, which is recommended in MPEG-4 [17]. The LPE padding scheme can be summarized as follows:

1. Determine the mean value m of all pixels within the object region in the boundary block.

2. The background region of the boundary block is filled with the mean value m obtained in Step 1.

3. Apply low-pass filtering to each pixel outside the object region in a recursive manner according to Eq. 1, beginning from the top-left corner of the boundary block and processing row by row to the bottom.

$$b'(i,j) = \frac{b(i-1,j) + b(i,j-1) + b(i,j+1) + b(i+1,j)}{4}, \tag{1}$$

   where $b(i,j)$ is the pixel value at a position (i,j) of the boundary block and $b'(i,j)$ is the pixel value after applying the low-pass filtering.

The coding efficiency of the LPE algorithm is higher than that of the repetitive padding scheme but is still poor [15]. This is because the LPE algorithm usually does not take the number of DCT coefficients that are exactly the same as the number of pixels within the original boundary block. Therefore, the method for the arbitrary shaped coding should be further refined.

In this study, the ***boundary block-searching*** (BBS) algorithm is proposed for the arbitrary shaped coding. The BBS algorithm is simple in its concept and significantly enhances the coding efficiency of the conventional LPE coding method. The key idea is to employ the strong correlation between boundary blocks even though they are not connected [20] in order to reduce the number of boundary blocks to be coded by the use of DCT. For an input boundary block, the most similar block (only considering the object region) is sought from the previously encoded data by the use of the BBS algorithm. Then, the boundary block is encoded as a position vector (pointing to the most similar block) instead of the DCT coefficients to achieve the low bit rate.

The rest of this paper are organized as follows. Section 2 describes the proposed BBS algorithm in detail. Then, simulation results are provided in Section 3. Finally, Section 4 briefly concludes this paper.

2 Boundary Block-Searching (BBS) Algorithm

2.1 Definitions

Figure 1 illustrates some definitions used in the proposed BBS algorithm. Assume that the size of the image block is 8×8. Some terms used in the proposed BBS algorithm are defined as follows:
1) *Search Area*: The search area is the set of dashed blocks shown in Fig. 1. It contains the previously reconstructed pixels with the predefined size determined by the parameter $W = 2^n \times 2^n$.
2) *Current Block*: The current block is a boundary block that is being encoded.
3) *Search Block*: Search blocks can be obtained by dividing the search area into overlapping blocks, in which the neighboring blocks are spaced at one-pixel interval in both horizontal and vertical directions. Search blocks have the same size as current blocks.
4) *Position Vector*: Each of search blocks in the search area needs a vector called *position vector* to denote its corresponding x- and y-direction positions in the search area and this vector is encoded as a $2n$-bit bitstream.

In image coding, image blocks are usually small (typically 8×8). Thus there exists strong correlation among image blocks [20], especially for the neighboring blocks. Therefore, the idea of the proposed BBS algorithm is to find the most similar block in other area of the image. Only the previous encoded data are useful to achieve this goal because the data behind the current block are not available in the decoder. The search area covers many previously encoded data and the overlapping concept can produce a large pattern area to seek a block that is the most similar to the current block. Thus the most similar block can be used as the prediction of the current block, and only $2n$ bits are required to be recorded. Therefore, the bit rate can be reduced significantly.

2.2 Search Area

Now, the problem is how to determine an adequate search area according to the boundary direction of the object. Boundary pixels that are neighboring to the background region in a boundary block can form a *boundary chain* and then can be used to predict the boundary direction. Here, the chain code [21] scheme is used to describe the direction with a given orientation shown in Fig. 2(a). According to the boundary direction, four cases of the search area are defined in Fig. 3 and the extra two bits called *searching-area* bit are required to identify which case the search area belongs to. The procedures of determining the search area are described as follows.

1. Find the boundary chain according to a given orientation and its corresponding chain code shown in Fig. 2(a).

2. Find the sum C of all codes shown in Fig. 2(b). If the sum C only contains a real number or the character H, go to Step 4. Otherwise, go to next step.

3. If the sum C contains a real number and the character H, we only keep the real number and go to next step.

4. The search area is set as that shown in Case 1 and Case 2 when the sum C is greater and smaller than zero, respectively. If the sum C is equal to zero, the search area is set as that shown in Case 3. Otherwise, the Case 4 is applied.

Once the search area has been determined, the most similar block (only for the object region) called *Prediction Boundary* (PB) block is sought in this area. The PB block is the most similar to the current block only for the object region and is used as the prediction of the current block. A corresponding criterion is the distortion D between the current and PB blocks, which is defined by

$$D = \sqrt{\Sigma_{i=1}^{n}(x_i - x_{pbi})^2/n}, \tag{2}$$

where n is the number of object pixels in the current block, and x_i and x_{pbi} is the object pixels in the current and PB blocks, respectively. If the distortion D is less than a predefined threshold T, the PB block is regarded as the prediction of the current block and it will be represented as a position vector only. In some cases, the PB block is very different from the current block. That is, the distortion D is larger than the predefined threshold T. In this case, the PB block cannot be used as the prediction of the current block and the conventional LPE or SADCT algorithms can be employed to encode the current block.

2.3 Sub-pixel Search

Basically, the search area is generated with the integer-pixel resolution. However, it is also possible to employ the sub-pixel concept to make the search area with the finer resolution. To achieve this goal, the search area is interpolated by a factor of two in both horizontal and vertical directions. Thus larger computational complexity and storage memory are required. Therefore, we only search the eight neighboring blocks with the sub-pixel resolution. Instead of fully searching the sub-pixel search area shown in Fig. 4. The steps of sub-pixel searching are described shown as follows.

1. Search the PB block with the integer-pixel resolution and record its corresponding distortion value.

2. Search the eight neighboring blocks of the best integer pixel PB with the sub-pixel resolution. Compute the distortions of the eight blocks and compare the nine distortion values (include the integer-pixel block). Then, the block that has the minimum distortion value as the PB block is selected.

Figure 4 shows the integral pixel $f(i,j)$ and its corresponding eight sub-pixels $S_0, \cdots, S_7$. The sub-pixels are estimated by interpolating the integral pixels and the sub-pixels S_0 and S_1 can be obtained by Eqs. 3 and 4, respectively. That is,

$$S_0 = \frac{f(i,j) + f(i,j-1) + f(i-1,j-1) + f(i-1,j)}{4} \tag{3}$$

and

$$S_1 = \frac{S_0 + f(i,j) + S_2 + f(i,j-1)}{4}. \tag{4}$$

The sub-pixels S_2, S_5 and S_7 can be calculated by the similar way as that for the sub-pixel S_0; while the sub-pixels S_3, S_4 and S_6 can be calculated by the similar way as that for the sub-pixel S_1.

2.4 Summarization

Figure 5 illustrates the flow chart of the proposed BBS algorithm. The encoding and decoding processes are summarized as follows:
A. Encoding

1. Given an input boundary block, find its corresponding search area according to the chain-code rule shown in Fig. 2(a) and record its corresponding search-area bit.

2. Search the PB block with the integer-pixel resolution in the search area by calculating the distortion D defined in Eq. 2.

3. If the distortion D is less than the predefined threshold T, calculate the distortion of eight neighboring blocks with the sub-pixel resolution. Otherwise, go to Step 6.

4. Compare nine distortion values of search blocks (include the integer-pixel block) and select the block that has the minimum distortion value as the new PB block.

5. If the PB block belongs to the original integer-pixel resolution, the current block is represented as a position vector and encoded as a $2n$-bit bitstream following search-area bit and a flag "11" is used. Otherwise, go to next step.

6. If the PB block belongs to the sub-pixel resolution, the current block is represented as a position vector and encoded as a $(2n+3)$-bit bitstream following the search-area bit and a flag "10" are recorded. Then, go back to the Step 1 to process the next boundary block.

7. If the distortion D is greater than or equal to the predefined threshold T, the LPE algorithm is employed to encode the boundary block as a bitstream that follows a flag "0" to the decoder; then go back to Step 1.

B. Decoding

1. Read and check whether the first flag bit is "0" or not. If yes, the inverse DCT operation is employed to reconstruct the boundary block. Otherwise, go to Step 2.

2. Read the second flag bit and check whether it is "1" or not. If yes, recover the boundary block from its corresponding search area with the integer-pixel resolution according to the position vector. Otherwise, go to Step 3.

3. Recover the boundary block from its corresponding search area with the sub-pixel resolution according to the position vector.

The BBS algorithm obtains the prediction of an input boundary block by searching previously encoded data. In general, most of the boundary blocks are encoded by position vectors instead of the DCT coefficients. Thus the coding bit rate can be significantly reduced.

3 Simulation Results

In computer simulation, two well-known MPEG-4 video sequences in the CIF (352×288) format, *Miss_American* and *Weather* video sequences, are used. The search area is of size 16×16 ($W = 16$, $n = 4$) and the LPE padding algorithm is performed based on the normal 8×8 DCT operation. Figures 6 and 7 show original images and their corresponding shapes of *Miss_American* and *Weather* video sequences, respectively, which are the corresponding first VOs of the frame 1 in these standard video sequences.

Figure 8(a) shows the rate-PSNR result obtained by the BBS and the LPE algorithms for the comparison only in the first VO of the *Miss_American* video sequence. Clearly, the proposed BBS algorithm outperforms the LPE algorithm by, at most, 3.5 dB improvement when the threshold value T is 4 and the bit-rate is 0.5 bit per pixel (bpp). Similar results for the *Weather* video sequence is shown in Fig. 8(b). The coding results for frame 1 of the *Miss_American* and *Weather* video sequences are depicted in Figs. 9 and 10, respectively.

Figure 11(a) shows the rate-PSNR results for the *Miss_American* video sequence obtained by the BBS algorithm with different threshold values. For a given threshold value, a curve represents the rate-PSNR results that are obtained by varying the quantization table employed in the DCT-based compression scheme. The rate-PSNR curves generated by the proposed BBS algorithm depend on the threshold value T. The larger threshold value T is used, the better PSNR result compared with that of the LPE algorithm can be obtained at the lower bit-rate range ($0.2 - 0.5$ bpp). Here most of the boundary blocks are coded by the proposed BBS algorithm, which has better performance than that of DCT-based coding results in low bit rates. On the contrary, the better PSNR results are obtained at the higher bit-rate range (> 0.5 bpp) when the smaller threshold value T is used. Under this condition, more boundary blocks are coded by the use of LPE algorithm, which results in better performance at higher bit rates. Similar results for the *Weather* video sequence can be observed in Fig. 11(b).

Figures 12 shows the ratio of boundary blocks encoded by the proposed BBS algorithm to all boundary blocks for the *Miss_American* and *Weather* video sequences. When a larger threshold value $T = 8$ is used, more boundary blocks are coded by the BBS technique. In this case, the poorer PSNR result compared with the LPE algorithm can be obtained in the higher bit-rate because the DCT operation will yield the higher PSNR value than that of the BBS algorithm. On the contrary, the DCT operation yields the poor PSNR result than that of the BBS algorithm in lower bit rates because that the PSNR result obtained from the DCT operation will be dramatically decreased when the bit rate is small. The proposed BBS algorithm still can preserve the PSNR quality by the use of a fixed threshold value T. When a smaller threshold value $T = 4$ is used, the PSNR results of the proposed BBS algorithm are 1–2 dB lower than that for the threshold value $T = 8$. However, the best PSNR results are obtained at high bit rates. For different applications, a suitable choice for threshold value T is required to yield a better rate-PSNR tradeoff.

4 Conclusion

Arbitrary shaped coding is an important tool in the object-based image coding. In this study, the proposed BBS algorithm can enhance the coding efficiency of the conventional

LPE padding technique. The coding efficiency gain is obtained by reducing the number of the boundary blocks that are coded by DCT. The strong correlation between the boundary blocks is employed to achieve this goal. Experimental results show that the BBS algorithm outperforms the LPE technique by, at least, 3 dB when the bit rates are within the range from 0.2 to 0.5 bpp for *Miss_American* video sequence. In the high bit-rate application, the PSNR gain of the proposed BBS algorithm can be even higher when choosing a suitable threshold T.

Acknowledgment

This research was partly supported by the National Science Council, Taiwan under contracts NSC-92-2213-E-224-047.

References

[1] ISO/IEC JTC1/SC29/WG11 N4668, Coding of moving pictures and audio, "Overview of the MPEG-4 Standard," Mar, 2002.

[2] ITU_T Recommendation H.261, "Video codec for audiovisual services at $p \times 64$ kbit/s," 1990.

[3] ISO/IEC 11172-2 MPEG-1 Video Coding Standard, "Information technology–Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s–Part 2: Video," 1993.

[4] ISO/IEC 13818-2 MPEG-2 Video Coding Standard, "Information technology–Generic coding of moving pictures and associated audio–Part 2: Video," 1995.

[5] ITU_T Recommendation H.263, "Video coding for low bitrate communication," 1996.

[6] C. S. Boon, J. Takahashi, and S. Kadono, "Experiment results of content-based texture coding," ISO/IEC JTC1/SC29/WG11 Doc. MPEG96/1029, 1996.

[7] C. S. Park, D. D. Hwang, J. T. Lim, K. H. Chang and S. D. Kim, "Daewoo proposal for region texture coding," ISO/IEC JTC1/SC29/WG11 MPEG95/m055, 1996.

[8] L. Chiariglione, "MPEG and multimedia communications," *IEEE Transactions on Circuit and System on Video Technology,* vol. 7, no. 1, pp. 5–18, Feb. 1997.

[9] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Transactions on Circuit and System on Video Technology,* vol. 7, no. 1, pp. 19–31, Feb. 1997.

[10] S. F. Chang and D. G. Messerschmitt, "Transform coding of arbitrary-shaped image segments," *ACM Multimedia,* vol. 1, pp. 83–90, 1993.

[11] A. Kaup, "Adaptive low-pass extrapolation for object-based texture coding of moving video," *SPIE Proceeding in Image Processing on Visual Communication,* vol. 3024, pp. 731–741, 1997.

[12] A. Kaup, "Object-based texture coding of moving video in MPEG-4," *IEEE Transactions on Circuit and System for Video Technology,* vol. 9, no. 1, pp. 5–15, Feb. 1999.

[13] J. H. Moon, J. H. Kweon and H. K. Kim, "Boundary block-merging (BBM) technique for efficient texture coding of arbitrary shaped object," *IEEE Transactions on Circuit and System for Video Technology,* vol. 9, no. 1, pp. 35–43, Feb. 1999.

[14] H. S. Kang, S. D. Kim and S. W. Lee, "Object based fractal coding," *Optical Engineering,* vol. 38, no. 12, pp. 2029–2040, Dec. 1999.

[15] G. Shen, B. Zeng and M. L. Liou, "Arbitrarily shaped transform coding based on a new padding techniques," *IEEE Transactions on Circuit and System for Video Technology,* vol. 11, no. 1, pp. 67–79, Jan. 2001.

[16] W. G. Chen, C. Gu and M. C. Lee, "Repetitive and morphological padding for object-based video coding," *IEEE International Conferences on Image Processing,* vol. 1, pp. 373–376, 1997.

[17] ISO/IEC JTC1/SC29/WG11, "MPEG-4 video verification model version 11.0," Doc. N2172, Mar. 1998.

[18] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 5, no. 1, pp. 59–62, Feb. 1995

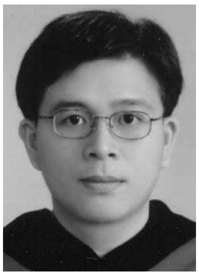
[19] T. Sikora, S. Bauer, and B. Makai, "Efficiency of shape-adaptive 2-D transforms for coding of arbitrarily shaped image segments,"*IEEE Transactions on Circuits and Systems for Video Technology,* vol. 5, no. 3, pp. 254–258, June 1995

[20] J. C. Tsai, C. H. Hsieh and T. C. Hsu, "A new dynamic finite-state vector quantization algorithm for image compression," *IEEE Transactions on Image Processing,* vol. 9, no. 11, pp. 1825–1836, Nov. 2000.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Second Edition, Chapter 11, pp. 644–646, Prentice Hall, New Jersey, 2002.

Hsuan T. Chang received his B.S. degree in electronic engineering from the National Taiwan Institute of Technology, in 1991, and M.S. and Ph.D. degrees in electrical engineering from National Chung Cheng University, Taiwan, in 1993 and 1997, respectively. He was a visiting researcher at the Laboratory for Excellence in Optical Data Processing, Department of Electrical and Computer Engineering, Carnegie Mellon University, from 1995 to 1996. He was an assistant professor in the Department of Electronic Engineering in Cheinkuo Institute of Technology, Changhua, Taiwan, from 1997 to 1999, an assistant professor in the Department of Information Management, Chaoyang University of Technology, Wufeng, Taiwan, from 1999 to 2001, and an assistant professor in the Department of Electrical Engineering of National Yunlin University of Science and Technology, Douliu, Taiwan, from 2001 to 2002, where he currently is an associate professor. His interests include image/video processing, optical information processing/computing, and bioinformatics. He has published more than 90 journal and conference papers. He served as the reviewer of several international journals and conferences and on the program committee in domestic and international conferences.

Dr. Chang is a member of *International Society for Optical Engineering* (SPIE), *Optical Society of America* (OSA), *Institute of Electrical and Electronic Engineers* (IEEE), International Who's Who (IWW), and *The Chinese Image Processing and Pattern Recognition Society.*



Chia-H. Yeh was born in Taiwan, on January 10, 1975. He received the B.S. and Ph.D. degrees from National Chung Cheng University (CCU), Chiayi, Taiwan, in 1997 and 2002, respectively, all from the Department of Electrical Engineering. Dr. Yeh received postdoctoral research fellowship award from Prof. C.-C. Jay Kuo's research group at the Department of Electrical Engineering-Systems, Viterbi School of Engineering, University of Southern California (USC) from August 2002 to December 2004. He was a technical consultant of ALi cooperation from August 2003 to August 2004. In the summer of 2004, he joined MAVs Lab. Inc. in the U.S. and has been the vice president and chief technology officer in MAVs Lab. Inc., Taiwan, since 2005.

His research interests are bioinformatics, image/audio/video signal processing, image/video database management, and optical information processing/computing. He has served as a reviewer for international journals and conferences and was an invited speaker at conferences. In 2002, he received the Outstanding Student Award from National Chung Cheng University. Dr. Yeh is a member of the Institute of Electrical and Electronics Engineers (IEEE) and International Society for Optical Engineering (SPIE).

Figure Captions:

Figure 1: Definitions used in the proposed BBS algorithm.
Figure 2: Chain code in eight connectivities and an example for finding the sum C of a boundary chain.
Figure 3: Four cases for selecting the search area according to the sum C of chain code.
Figure 4: Estimate of sub-pixels by the use of interpolation.
Figure 5: Flowchart of the proposed BBS algorithm.
Figure 6: First VO of the *Miss_American* video sequence: (a) video object and (b) the corresponding shape.
Figure 7: First VO of the *Whether* video sequence: (a) video object and (b) the corresponding shape.
Figure 8: Rate-PSNR results of boundary block coding using the proposed BBS and LPE algorithms for the (a) *Miss_American* and (b) *Weather* video sequences.
Figure 9: Frame 1 of the *Miss_American* video sequence: (a) the original image and (b) the image coded by the proposed BBS algorithm with the threshold value $T = 4$ (bit rate = 0.5 bpp and PSNR= 37.8 dB).
Figure 10: Frame 1 of the *Weather* video sequence: (a) the original image and (b) the image coded by the proposed BBS algorithm with the threshold value $T = 4$ (bit rate = 1 bpp and PSNR= 31.8 dB).
Figure 11: Rate-PSNR results under different threshold values using the proposed BBS algorithm for the (a) *Miss_American* and (b) *Weather* video sequences.
Figure 12: Ratio comparison of the boundary blocks coded by the proposed BBS algorithm to all boundary blocks for the *Miss_American* and *Weather* video sequences.

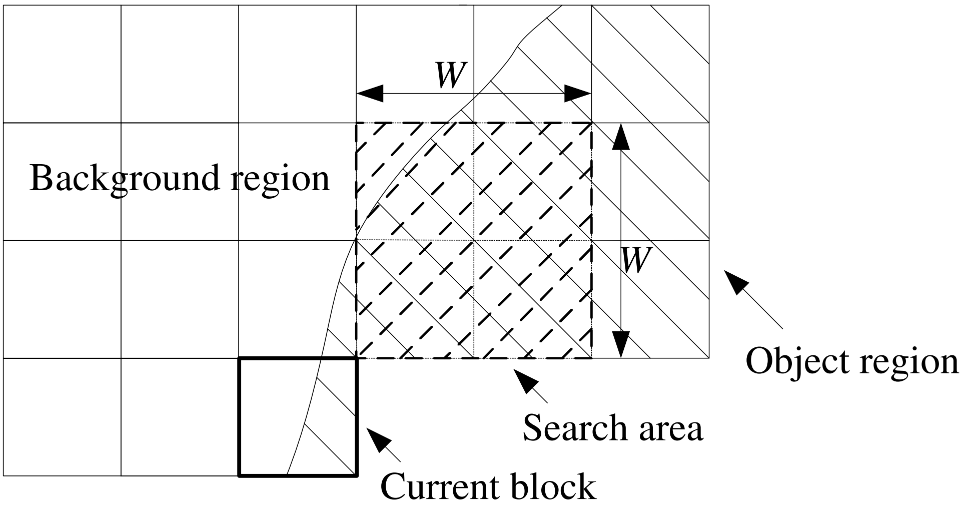
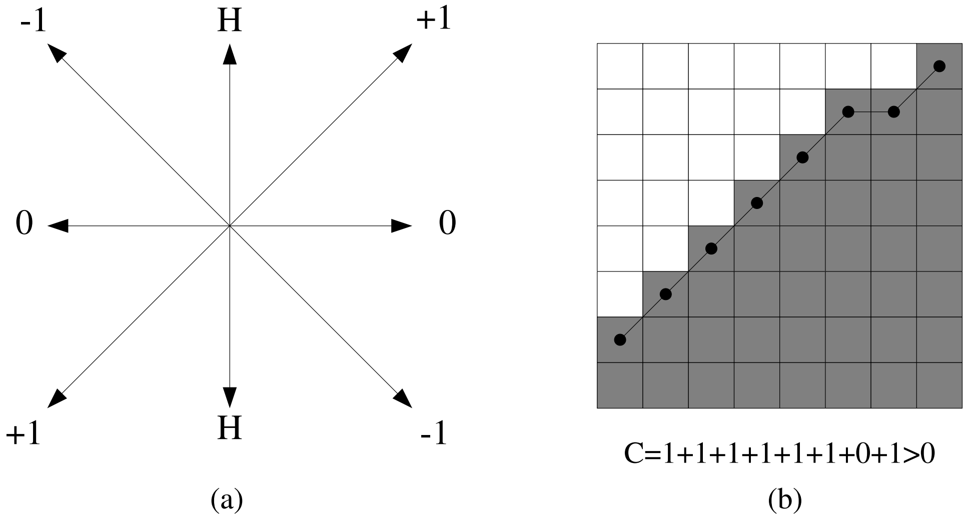


Figure 1: Definitions used in the proposed BBS algorithm.

.



(a)

(b)

Figure 2: Chain code in eight connectivities and an example for finding the sum C of a boundary chain.

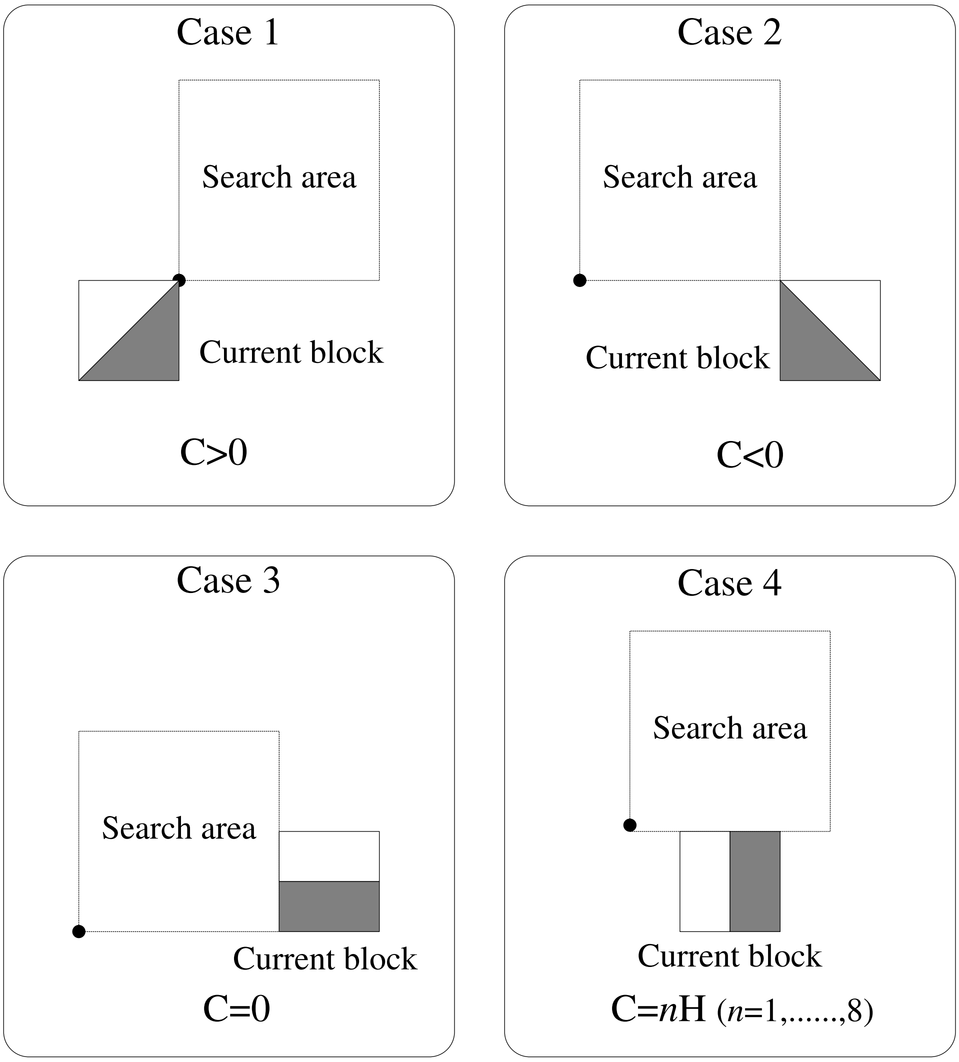


Figure 3: Four cases for selecting the search area according to the sum C of chain code.

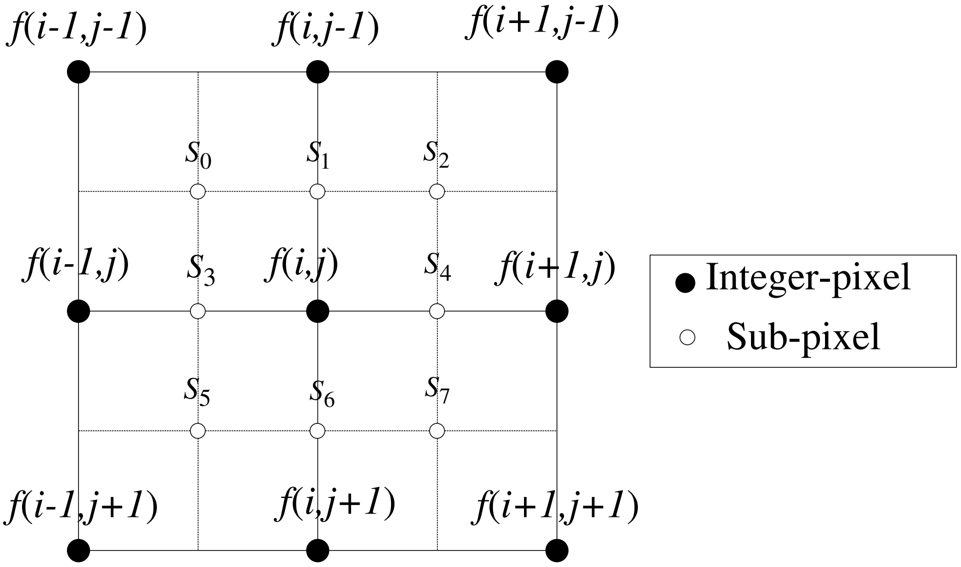


Figure 4: Estimate of sub-pixels by the use of interpolation.

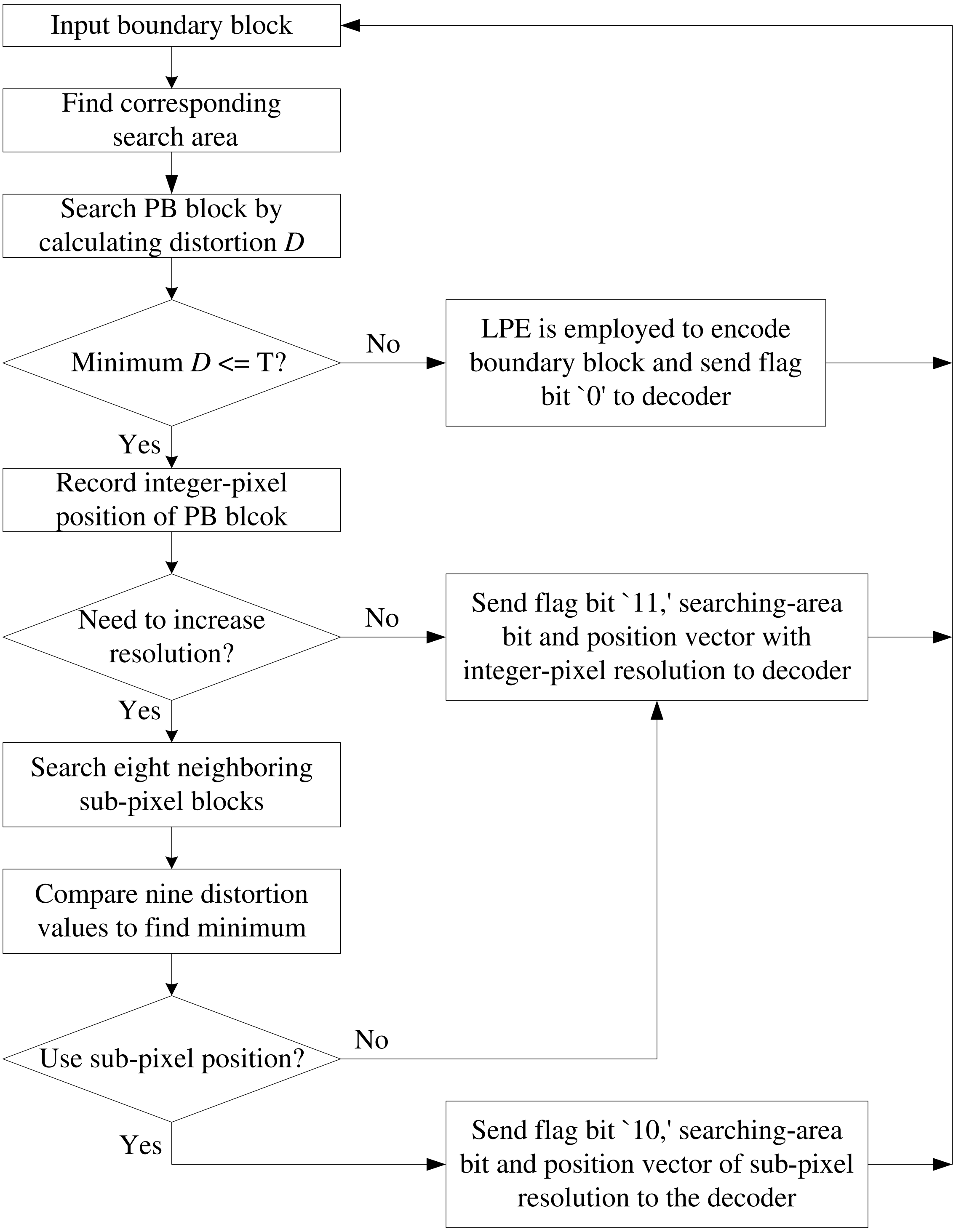
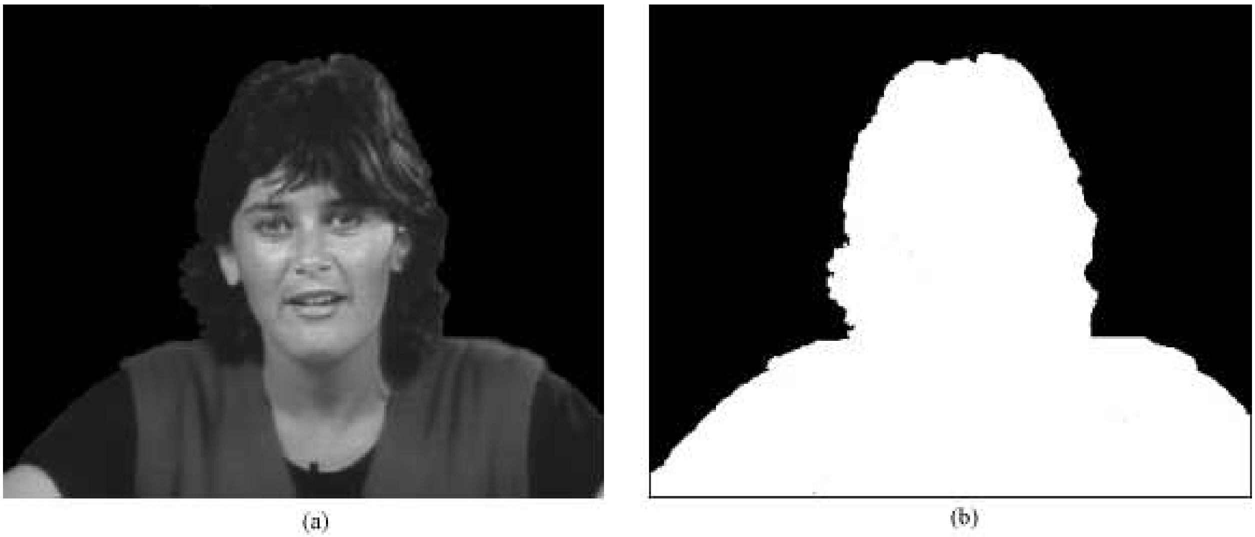

Input boundary block
Find corresponding search area
Search PB block by calculating distortion *D*
Minimum *D* <= T?
No
LPE is employed to encode boundary block and send flag bit `0' to decoder
Yes
Record integer-pixel position of PB blcok
Need to increase resolution?
No
Send flag bit `11,' searching-area bit and position vector with integer-pixel resolution to decoder
Yes
Search eight neighboring sub-pixel blocks
Compare nine distortion values to find minimum
Use sub-pixel position?
No
Yes
Send flag bit `10,' searching-area bit and position vector of sub-pixel resolution to the decoder

Figure 5: Flowchart of the proposed BBS algorithm.



(a)

(b)

Figure 6: First VO of the *Miss_American* video sequence: (a) video object and (b) the corresponding shape.

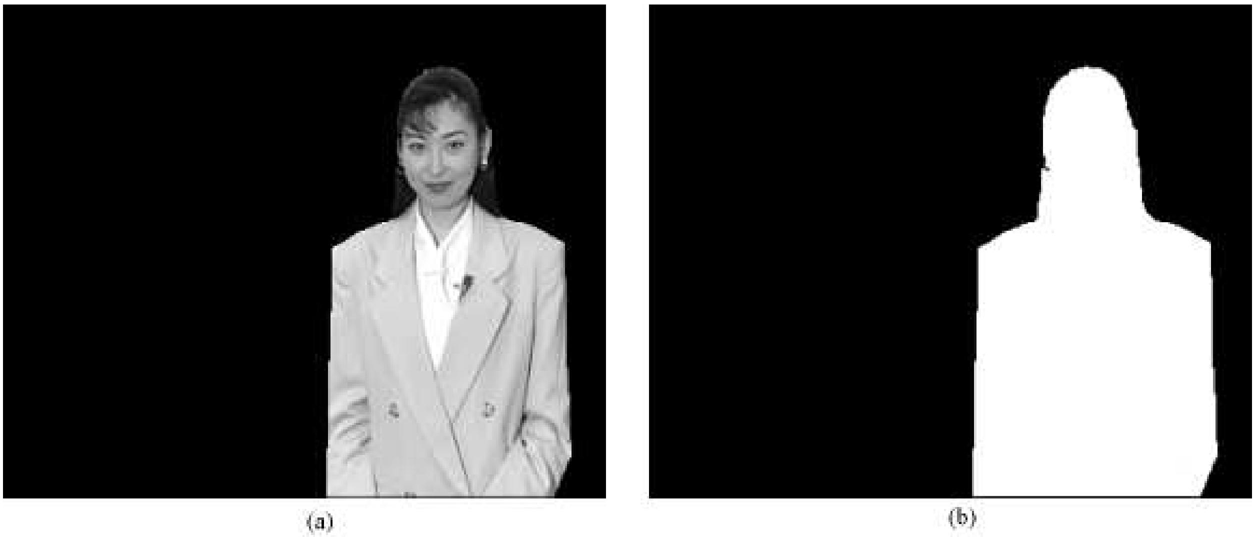
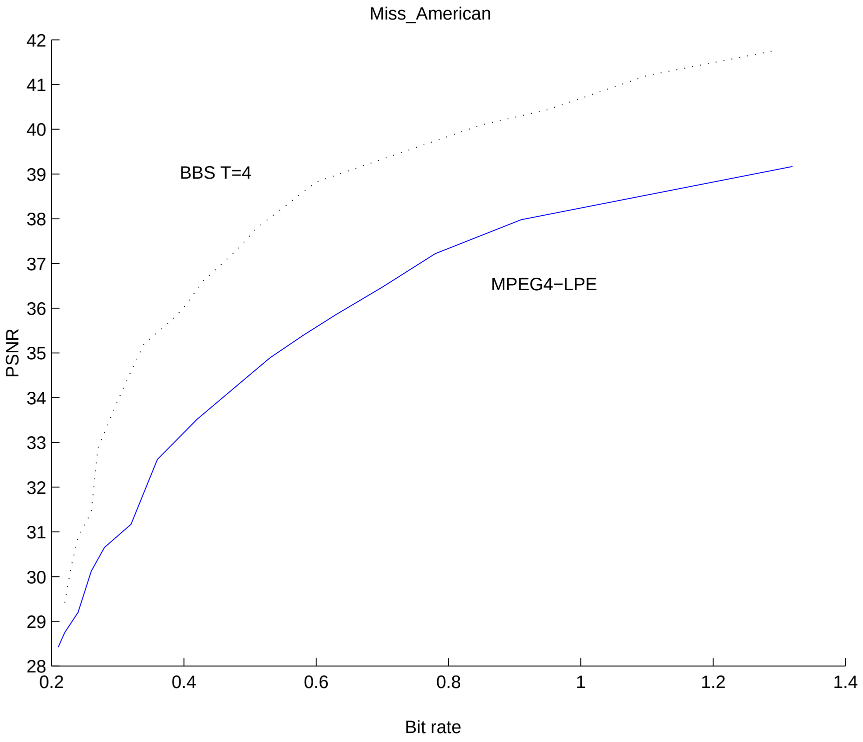
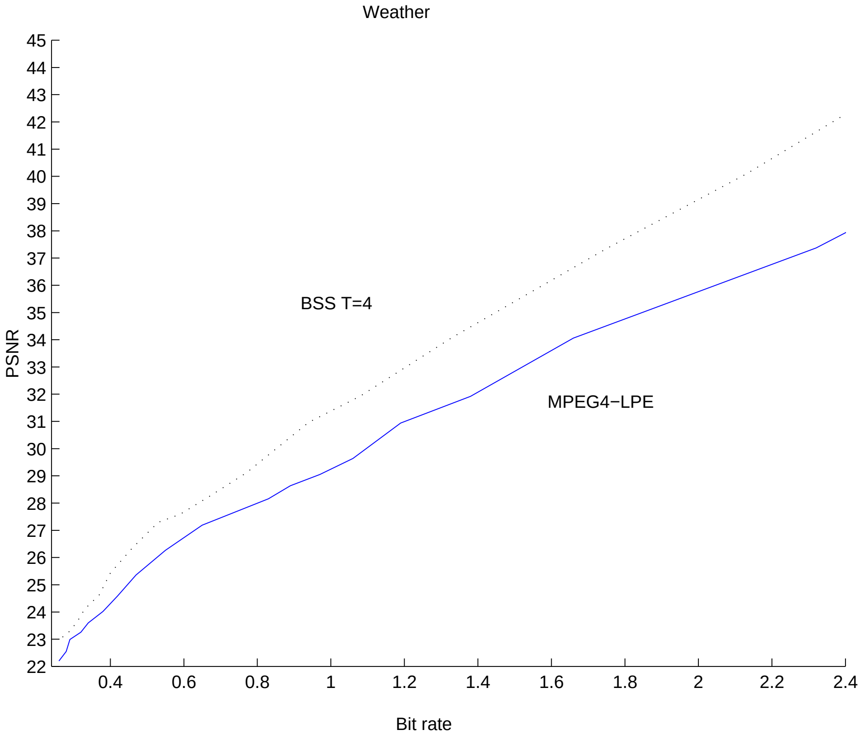


Figure 7: First VO of the *Whether* video sequence: (a) video object and (b) the corresponding shape.



(a)



(b)

Figure 8: Rate-PSNR results of boundary block coding using the proposed BBS and LPE algorithms for the (a) *Miss_American* and (b) *Weather* video sequences.

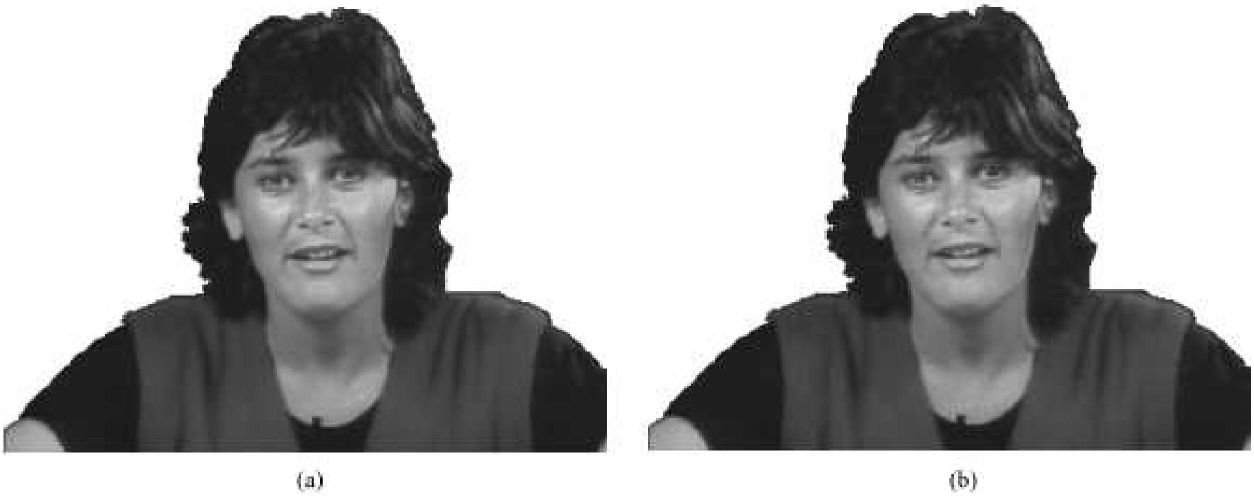


Figure 9: Frame 1 of the *Miss_American* video sequence: (a) the original image and (b) the image coded by the proposed BBS algorithm with the threshold value $T = 4$ (bit rate = 0.5 bpp and PSNR= 37.8 dB).

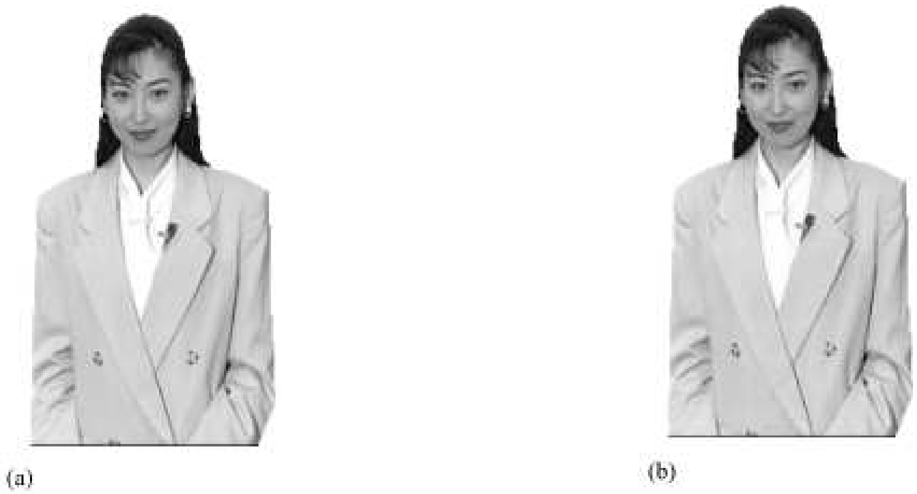
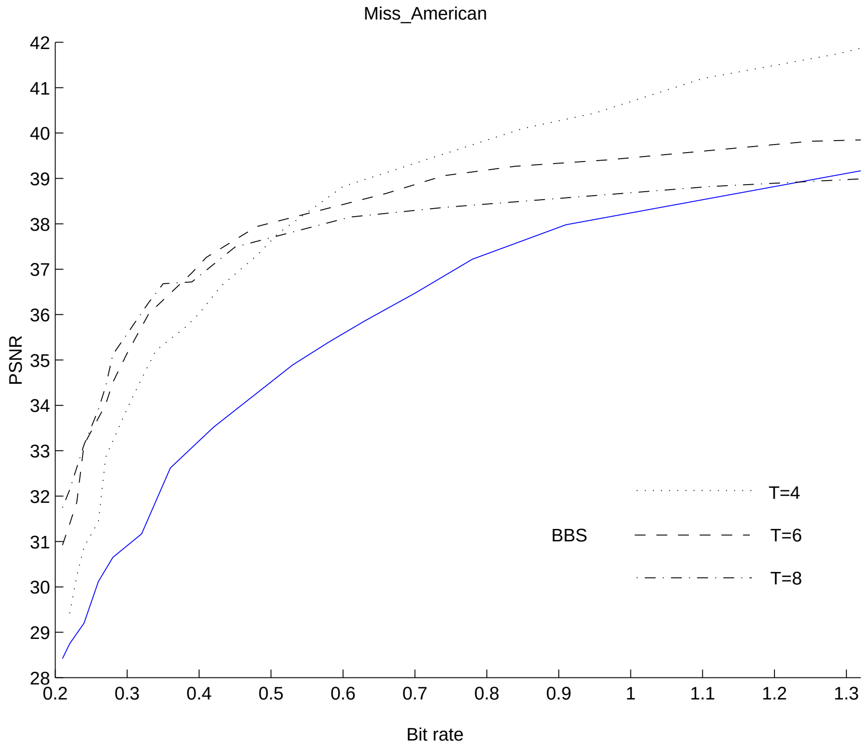
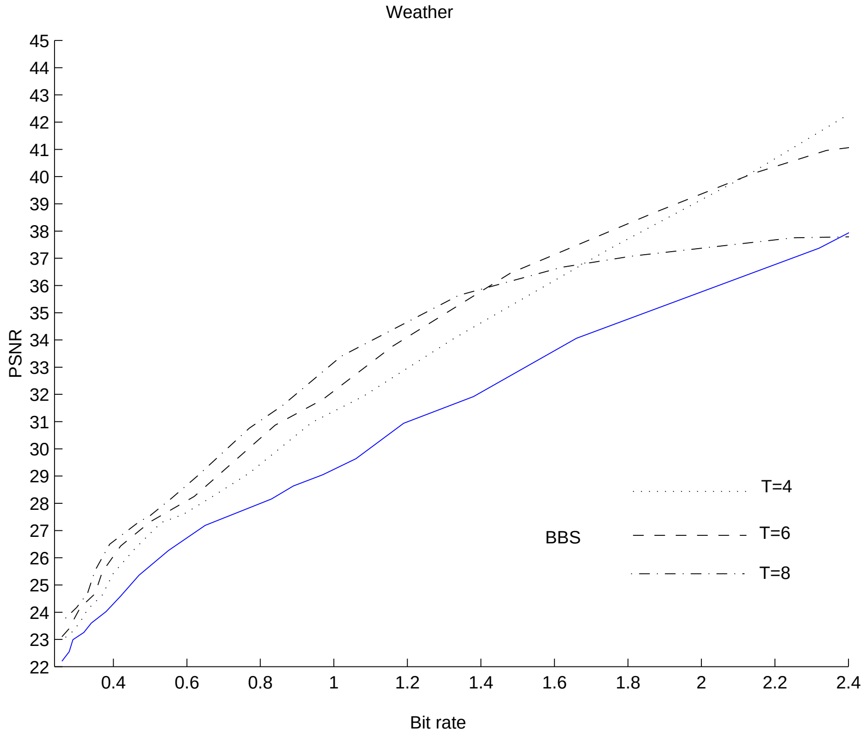


Figure 10: Frame 1 of the *Weather* video sequence: (a) the original image and (b) the image coded by the proposed BBS algorithm with the threshold value $T = 4$ (bit rate = 1 bpp and PSNR= 31.8 dB).



(a)



(b)

Figure 11: Rate-PSNR results under different threshold values using the proposed BBS algorithm for the (a) *Miss_American* and (b) *Weather* video sequences.

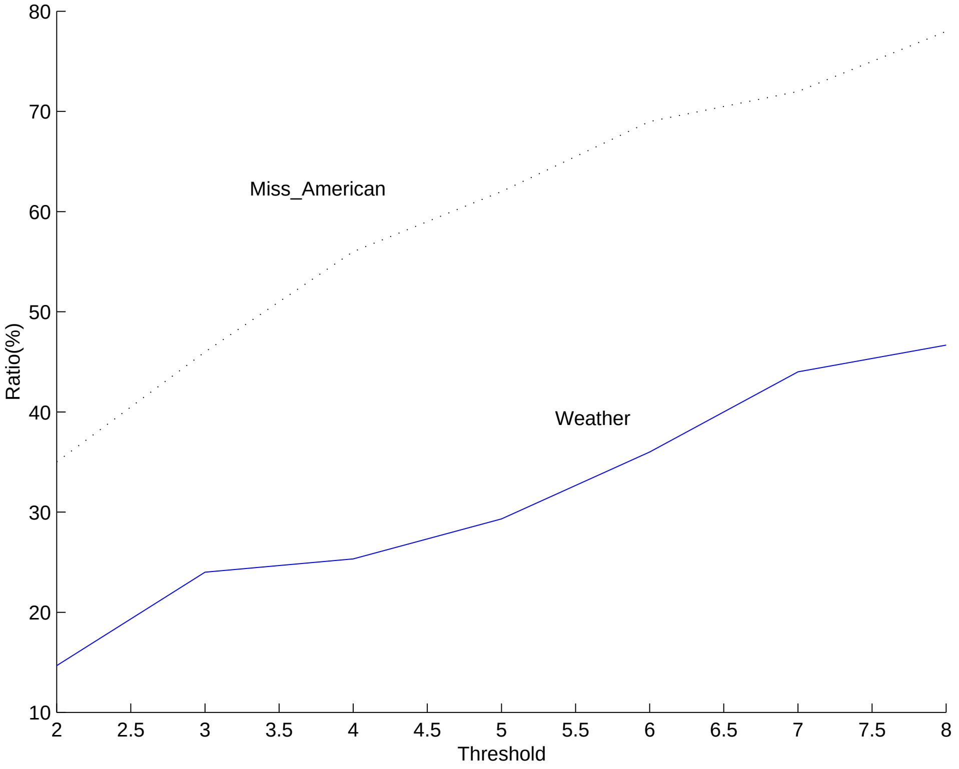


Figure 12: Ratio comparison of the boundary blocks coded by the proposed BBS algorithm to all boundary blocks for the *Miss_American* and *Weather* video sequences.