Transactions Letters

Color Quantization of Compressed Video Sequences

Wan-Fung Cheung and Yuk-Hee Chan

***Abstract*—This paper presents a novel color quantization algorithm for compressed video data. The proposed algorithm extracts discrete cosine (DC) transform coefficients and motion vectors of blocks in a shot to estimate a cumulative color histogram of the shot and, based on the estimated histogram, design a color palette for displaying the video sequence in the shot. It significantly reduces the complexity of the generation of a palette by effectively reducing the number of training vectors used in training a palette without sacrificing the quality. The palette obtained can provide a good display quality even if zooming and panning exist in a shot. The experimental results show that the proposed method can achieve a significant signal-to-noise ratio improvement as compared with conventional video color-quantization schemes when zooming and panning are encountered in a shot.**

***Index Terms*—Color display, color palette, color quantization, compressed domain video processing, compressed video, cumulative color histogram, DC sequence, MPEG.**

I. Introduction

MANY currently available image-display devices can only display a limited number of colors simultaneously. Accordingly, in order to display a full-color video sequence, color quantization [1] is necessary for a display device to produce a set of a limited number of colors based on the contents of the sequence and then use it to represent all the colors appearing in the sequence. The set of limited colors is generally called a palette.

Either a fixed palette or an adaptive palette can be used to display a video sequence. The implementation of the former approach is easy, but the output quality is generally poor. As for the latter approach, there are many implementation schemes. One of the simplest ways is to produce a palette for each frame of the sequence. However, though its output provides the optimum distortion performance in terms of mean square error, this approach is seldom used, as the computational cost is very high and the frequent changing of color palette results in screen flicker. In practice, an alternative in which a palette is produced for each shot is used instead, as flicker between shots is usually not detectable by human eyes.

A number of color quantization schemes were proposed to generate a palette based on some selected colors called *training vectors* [1]–[4]. Examples include the median-cut algorithm [1], the octree quantization algorithm [2], and the variance-minimization algorithm [4]. The training vectors involved in the palette-generation process plays a significant role in the process. They are treated as the representatives of all colors to be handled. Hence, they should be unbiased and cannot be too few. However, the complexity of the process is exponentially proportional to the number of training vectors. A compromise has to be reached in order to produce a palette at a reasonable computation cost.

Tricks have been proposed to reduce the computational effort for producing a palette by reducing the number of training vectors carefully. For example, one can select a key frame from a shot and use the colors that appear in the key frame as training vectors. One can also use the so-called discrete cosine (DC) sequence of a shot or even the DC image of the key frame of a shot to generate training vectors for training purposes [5], [6]. Using key frames to generate color palettes can reduce training vectors significantly, but it does not work when the shot of interest involves zooming and panning. Some colors may be missing before the zoom but appear after the zoom. The colors of a key frame in such a shot cannot represent all colors appearing in the shot.

In the past, color palettes have been generated with uncompressed data. However, most video sequences nowadays come in compressed formats such as [7] and [8]. Hence, it would be attractive to have a simple algorithm which can produce a color palette from the compressed data directly, as it can save an amount of computation effort for video decompression.

In this paper, we propose a novel color palette design method for compressed video sequences. This method extracts DC coefficients and motion vectors from a MPEG bitstream to produce a color palette for a shot such that one can quantize a video sequence efficiently.

II. DC Sequence

Most current video coding standards such as MPEG divide frames in a sequence into I-, P-, and B-frames [8]. All frames are partitioned into a number of blocks of size 8×8 and each block is then either intracoded or intercoded. Specifically, all blocks in an I-frame are intracoded while blocks in a P- or B-frame can be either intracoded or intercoded.

Manuscript received May 7, 2002; revised September 27, 2002. This work was supported by a Grant (A046) from the Center for Multimedia Signal Processing, Hong Kong Polytechnic University. This paper was recommended by Associate Editor H. Gharavi.

The authors are with the Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, Hong Kong Special Administrative Region, China (e-mail: enyhchan@polyu.edu.hk).

Digital Object Identifier 10.1109/TCSVT.2003.809831

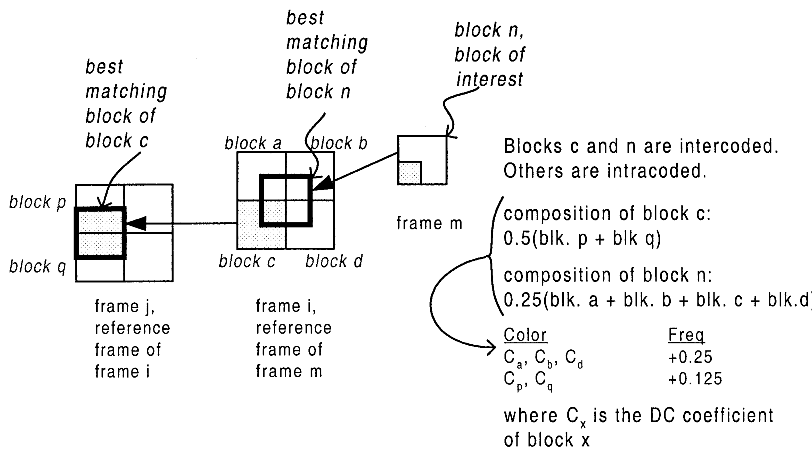


Fig. 1. Extracting colors of an intercoded block to update the cumulative histogram.

In [5], a DC image is defined as the image consisting of DC coefficients of all blocks in an image. A video sequence contains a number of frames and its corresponding DC sequence is defined accordingly to be a sequence of DC images associated with the frames. An encoded video sequence contains a number of blocks which are either intracoded or intercoded. Without decompressing the video sequence, it is impossible to get the DC sequence. A method was suggested in [5] to obtain an approximated DC sequence as follows. For an intracoded block, its DC coefficient is directly extracted from the bit-stream. As for an intercoded block, its best matching block in the reference frame is first identified with the associated motion vector of the block. Its DC coefficient is then estimated to be the weighted sum of the DC coefficients of the blocks which are partially covered by the best matching block in the reference frame. In formulation, we have

$$P(DC) = \sum_{i=1}^{4} P_{r,i}(DC) \times w_i \tag{1}$$

where $P(DC)$ denotes the DC coefficient of an intercoded block, $P_{r,i}(DC)$ denotes the DC coefficient of the ith block covered by the associated best matching block in the reference frame, and w_i is the corresponding area proportion being covered by the best matching block. After obtaining the approximated DC sequence of a shot, all DC coefficients are used as training vectors to generate a palette.

This method can effectively reduce a number of training vectors without ignoring any frame in a shot. Accordingly, it can provide an unbiased palette for the shot successfully. However, as it generates imaginary colors during the estimation of the DC coefficients of intercoded blocks, the number of training vectors can be as many as the total number of blocks in a shot and sometimes it is still very computation-intensive.

III. The Proposed Algorithm

In our proposed algorithm, a cumulative color histogram of the shot is estimated and the palette is generated based on the estimated cumulative histogram. A cumulative color histogram is the distribution of colors in a shot and it tells the frequency of occurrence of a color appearing in the shot. In our algorithm, the cumulative histogram is estimated in two steps. In the first step, all DC coefficients of intracoded blocks in a shot are considered as the representative colors of their blocks and extracted to construct a color histogram. In the second step, intercoded blocks are handled. For each intercoded block, its representative color is considered as a composite of some existing colors previously extracted from the intracoded blocks and the frequency of occurrence of a particular involved color is increased by its proportion in the composition. The proportion is determined based on the proportion of area overlapped by a block. Fig. 1 shows an example of how to extract colors of an intercoded block and update the cumulative histogram. In this example, all blocks other than blocks n and c are intracoded. As all intracoded blocks were handled in the first stage, representative colors of blocks a, b, d, p, and q, say, C_a, C_b, C_d, C_p, and C_q, were included in the histogram already. The best matching block of block n in the reference frame covers three intracoded blocks and one intercoded block the best matching block of which in the corresponding reference frame in turn covers two intracoded blocks. As a result, the frequency of occurrence of C_a, C_b, C_d, C_p, and C_q should be increased by 0.25, 0.25, 0.25, 0.125, and 0.125, respectively, in the cumulative histogram according to the area proportions of the blocks in the composition of block n.

In practice, the estimation of the cumulative histogram can be implemented in a GOP-by-GOP or even frame-by-frame manner as long as the block dependency is taken care of. By doing so, it is not necessary to process all intracoded blocks in a shot before handling an intercoded block, which can reduce the time to obtain the estimation result.

After handling all intercoded blocks in a shot, a cumulative color histogram is constructed. This histogram is different from a conventional color histogram in a way that the frequency of occurrence of a color could be a fractional number. Any color whose frequency of occurrence is larger than zero is considered as a representative color of the shot and used as one of the training vectors in the training process. The frequency of occurrence is taken into account during the training process.

TABLE I
TOTAL NUMBER OF DIFFERENT REPRESENTATIVE COLORS IN DIFFERENT SHOTS

Video Sequence	QP	Total no. of frames	Total no. of different representative colors: DC-sequence	Approximated DC-sequence	Ours
Negotiator Shot-1	16	144	82673	48064	6768
Negotiator Shot-2	16	144	82323	26868	6222
Negotiator Shot-3	8	115	46321	12368	3112
Negotiator Shot-4	12	201	68513	25841	11350
Vertical-Limit Shot-1	16	152	72179	43486	9378
Vertical-Limit Shot-2	8	68	47738	36242	8356

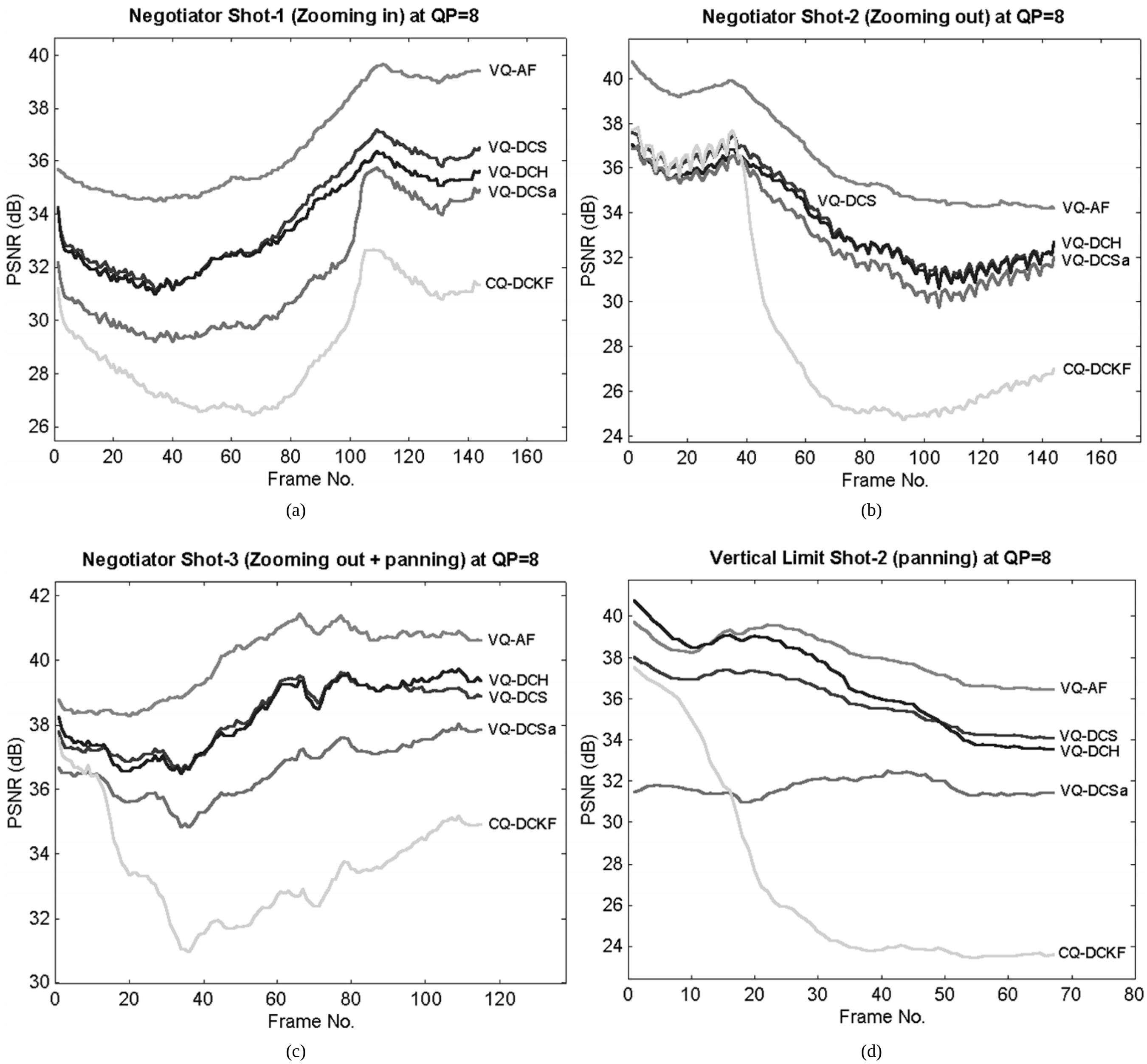


Fig. 2. PSNR performance of various palette generation schemes at QP = 8 under different conditions: (a) zooming in; (b) zooming out; (c) zooming out + panning; and (d) panning.

The proposed approach significantly reduces the number of training vectors used in color quantization. One can see that the maximum number of training vectors is equivalent to the total number of intracoded blocks in a shot. In practical applications, only a few of blocks in a shot are intracoded. Table I shows the total number of different representative colors involved in the generation of a color palette for a shot when different approaches are used. It shows that the proposed method can

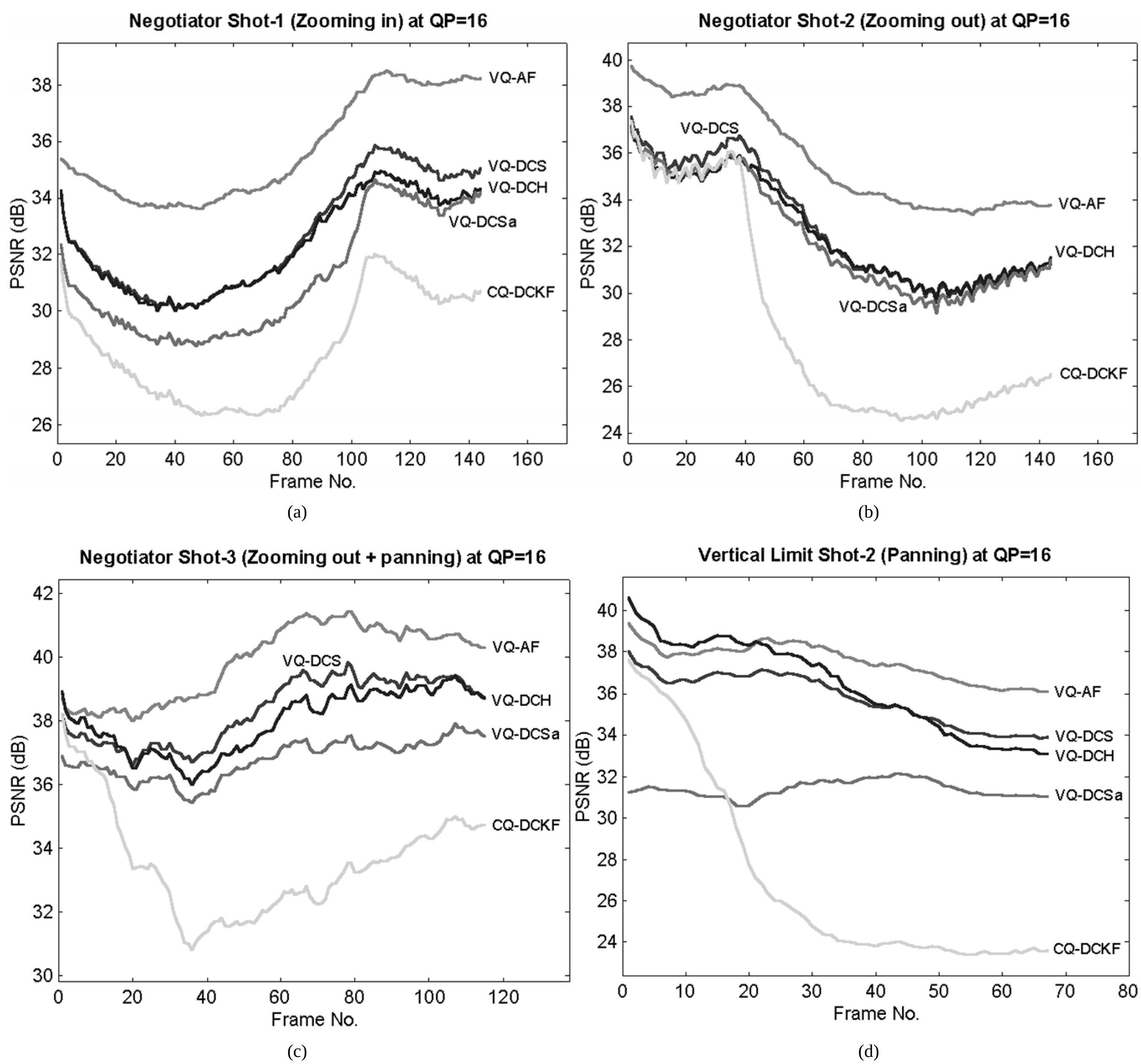


Fig. 3. SNR performance of various palette generation schemes at QP = 16 under different conditions: (a) zooming in; (b) zooming out; (c) zooming out + panning; and (d) panning.

reduce the total number of representative colors to 20%–40% of that required by the approach using an approximated DC sequence. Simulation results also shows that, when the proposed approach is used, only one third to one fifth of time is required to generate a color palette as compared with the approach using an approximated DC sequence.

IV. Simulation Results

Simulation has been carried out to evaluate the performance of the proposed approach. In our simulation, two movies in DVD format, *Negotiator* and *Vertical Limit*, were used as the source input of the simulation.

In our simulation, camera breaks were detected and the movies were split into a number of shots accordingly. Each shot was then compressed with an MPEG-1 video coder [7]. The ITU-T Recommendation H.263 [8] quantization method was adopted, and the motion search range was [−16, 15.5] during the compression. In order to simplify the testing process, only the first frame was encoded as I-frame and all other frames in a shot were encoded as P-frames with a fixed quantization parameter (QP). Color palettes of 256 colors each were then generated with various schemes. Specifically, the schemes we studied are as follows.

VQ-EF A color palette is generated for each frame in a shot with all colors in the frame.

TABLE II
AVERAGE PSNR PERFORMANCE OF VARIOUS COLOR QUANTIZATION SCHEMES

Video sequence	QP	Average PSNR per frame in a shot (dB)									
		UQ-EF	VQ-EF	VQ-AF	VQ-DCS	VQ-DCSa	VQ-DCH	CQ-DCKF	MC-DCS	MC-DCSa	MC-DCH
Negotiator Shot-1 (Zooming In)	8	23.29	37.80	36.65	33.92	31.65	33.52	28.94	32.70	28.16	32.50
	12	23.19	37.13	36.05	33.17	31.58	32.84	28.79	32.42	28.57	31.94
	16	23.07	36.71	35.70	32.67	31.15	32.35	28.66	31.81	28.67	31.51
Negotiator Shot-2 (Zooming Out)	8	23.35	37.93	36.67	33.90	33.07	33.66	29.17	33.50	30.05	32.62
	12	23.22	37.36	36.17	33.30	32.83	33.11	28.93	33.03	30.05	32.25
	16	23.16	36.93	35.82	32.90	32.41	32.69	28.67	32.39	28.98	32.02
Negotiator Shot-3 (Zooming Out + Panning)	8	24.02	41.11	40.01	38.28	36.61	38.30	33.58	37.82	34.95	37.65
	12	23.98	40.97	39.79	38.17	36.21	37.36	33.56	37.72	34.89	37.54
	16	23.95	40.97	39.88	38.34	36.80	37.98	33.54	37.52	35.05	37.50
Vertical Limit Shot-2 (Panning)	8	20.50	39.57	38.11	35.97	31.74	36.80	27.12	35.49	32.39	34.98
	12	20.48	39.41	37.68	35.77	31.70	36.56	27.09	35.63	32.22	35.90
	16	20.47	39.24	37.56	35.78	31.41	36.41	27.07	35.11	32.10	35.60

VQ-AF A single color palette is generated for a shot with all colors in the shot.

VQ-DCS A single color palette is generated for a shot with the DC sequence of the shot.

VQ-DCH A single color palette is generated for a shot with the proposed approach.

VQ-DCSa A single color palette is generated for a shot with the approximated DC sequence of the shot.

CQ-DCKF A single color palette is generated for a shot with the approach proposed in [6].

UQ-EF Fixed uniform quantization of R, G and B components, bit allocation = (3,3,2).

The prefix "VQ-" implies that the LBG algorithm [9] is used to generate a color palette with training colors in a particular scheme.

Figs. 2 and 3 show the peak signal-to-noise ratio (PSNR) performance of various schemes at different QPs in different cases. Here, PSNR is defined as

$$\mathrm{PSNR} = 10 \log \frac{255^2}{\mathrm{MSE}} \tag{2}$$

where MSE is the mean square error of the colors between a particular frame and its color-quantized output. Table II shows the average PSNR per frame in a shot for different schemes.

The output of UQ-EF suffers from very severe degradation and contouring. The simulation results of the proposed and other schemes are significantly better. VQ-EF provides an optimized PSNR performance as it can minimize the color quantization error for each frame. However, the complexity is very high. It is not only because of its frame-oriented nature, but the huge number of training colors involved in the training process of palette generation as well. Besides, decompression is necessary before extracting training colors. This makes it not applicable for real time applications. Moreover, frequent change of color palette leads to screen flicker.

All shot-oriented schemes can reduce screen flicker. Among them, at a cost of huge complexity that makes it impossible for real-time applications, VQ-AF provides the best PSNR performance as all colors in the shot are taken into account during its training process. It is used here as a reference for evaluating how close a particular shot-oriented scheme's performance is to the best. CQ-DCKF uses the DC image of a key frame to generate a color palette and hence its computational cost is very low. However, comparatively, it cannot handle cases such as zooming and panning well as, in such cases, the shot cannot be well represented with a single key frame. This fact is reflected in Figs. 2 and 3 that the PSNR of the output sequence gets worse and worse when the shot involves panning or zooming. The problem is that the size of the training color set is too small. The palette obtained is very biased to the key frame and hence the output quality is very sensitive to the scene change detection algorithm.

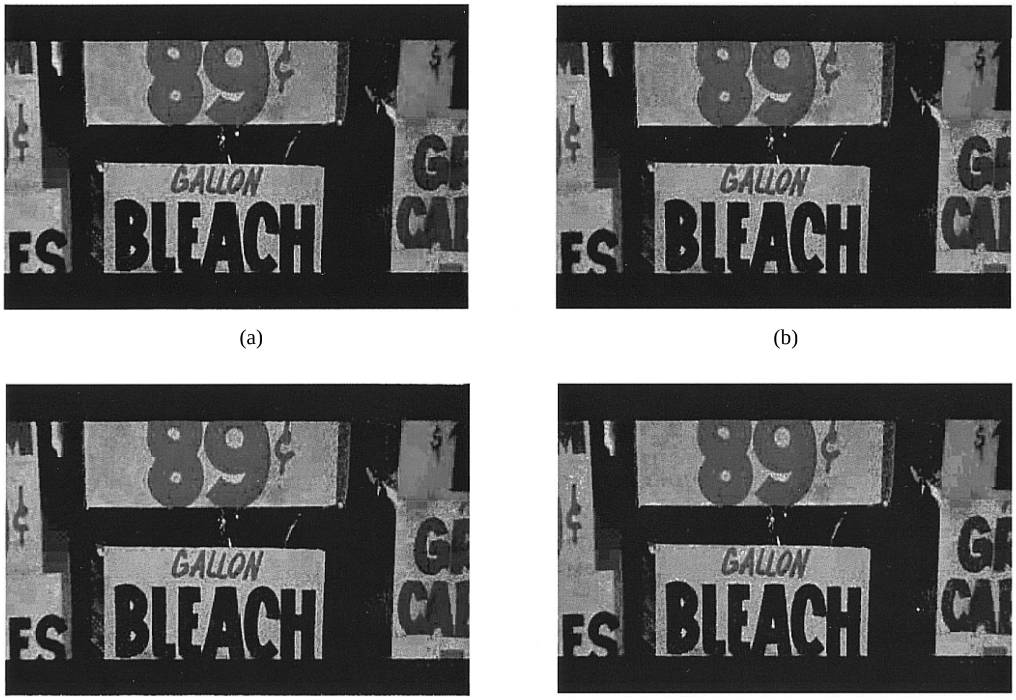
VQ-DCS, VQ-DCSa and the proposed VQ-DCH can handle zooming and panning cases. The proposed scheme is superior to the other two as its complexity is much lower. As shown in Table I, the number of training colors involved in the proposed scheme is much fewer. This superiority is not gained at a cost of quality. As shown in Figs. 2 and 3, the PSNR performance of the proposed scheme is better than that of VQ-DCSa and, as compared with that of VQ-DCSa, very close to that of VQ-DCS. The inferiority of VQ-DCSa might be due to the estimation error of the DC coefficients of intercoded blocks when VQ-DCSa is exploited. In general, the proposed method can, respectively, achieve a PSNR improvement of 4–9 dB and 1–5 dB as compared with CQ-DCKF and VQ-DCSa.

Figs. 4 and 5 show some simulation results for subjective evaluation. One can see that the output of the proposed scheme can report the colors more faithfully. This can be observed by comparing the orange and blue colors on the right of the pictures in Fig. 4. In Fig. 5, the results of VQ-DCS and CQ-DCKF are biased to blue while ours is not.

Using the cumulative color histogram estimated with the proposed approach is generally superior to using the DC sequence in extracting training colors for palette generation. The proposed algorithm can work with any palette generation algorithms besides the LBG algorithm to provide a good PSNR performance. Fig. 6 shows the case when the median-cut algorithm [1] is used. In this figure, the prefix "MC-" implies that the median-cut algorithm [1] is used to get the palette with the training vectors. The complexity of the median-cut algorithm is much lower than that of the LBG algorithm and real-time color quantization can be easily achieved with a typical Pentium-II computer.

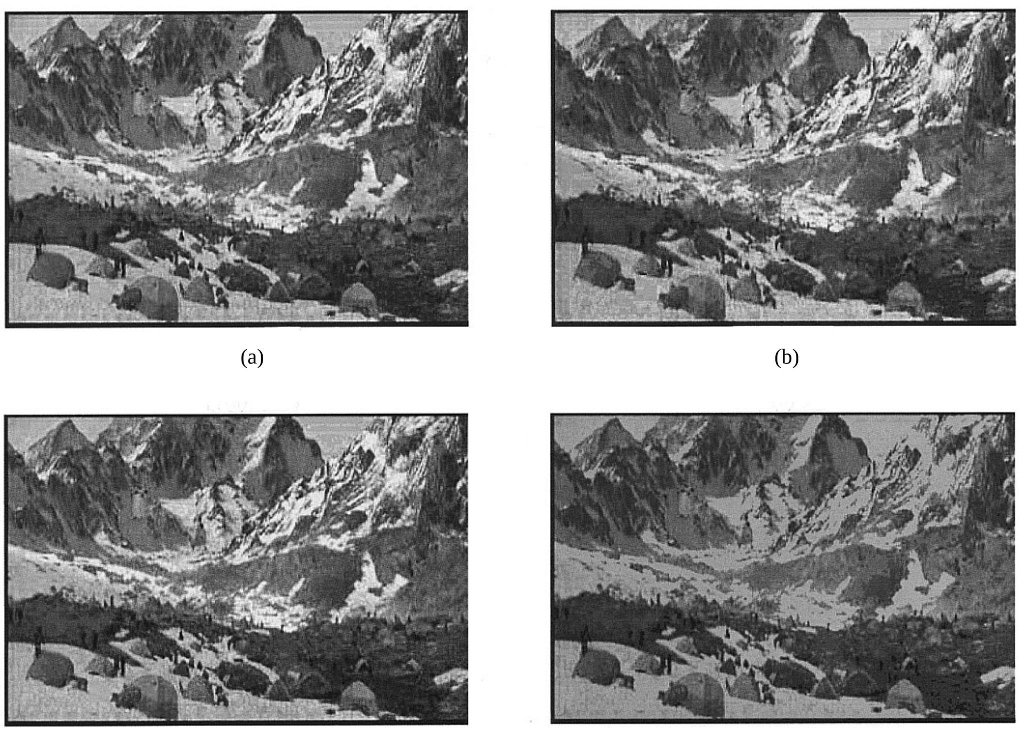
V. CONCLUSIONS

Limited color palette devices are popular nowadays. Color quantization has to be carried out so as to display true color video sequences with these devices. In this paper, a novel color quantization scheme for compressed video sequences is proposed. This scheme generates a color palette for each shot based on an estimated cumulative color histogram of the shot. It operates in the compressed domain directly and hence is able to save an amount of computation effort for decompression. It produces no screen flicker and can handle both zooming and



(a) (b) (c) (d)

Fig. 4. Performance of various color quantization schemes in zooming-in case: (a) original of the 90th frame of the MPEG-encoded *Negotiator* Shot-1 (QP16); (b) VQ-DCS output; (c) VQ-DCH output; and (d) CQ-DCKF output.



(a) (b) (c) (d)

Fig. 5. Performance of various color quantization schemes in panning case: (a) original of the 66th frame of the MPEG-encoded *Vertical Limit* Shot-2 (QP16); (b) VQ-DCS output; (c) VQ-DCH output; and (d) CQ-DCKF output.

panning cases. Unlike the approach used in [6], the proposed approach generates a color palette which is not biased to a single frame. As compared with the approach using DC sequences, the number of the training colors involved in the palette generation is significantly reduced, which in turn reduces the complexity of the process significantly. Simulation results show that the quality of the output obtained with the proposed scheme is better than those obtained with other conventional shot-oriented color quantization schemes which operate in the compressed domain.

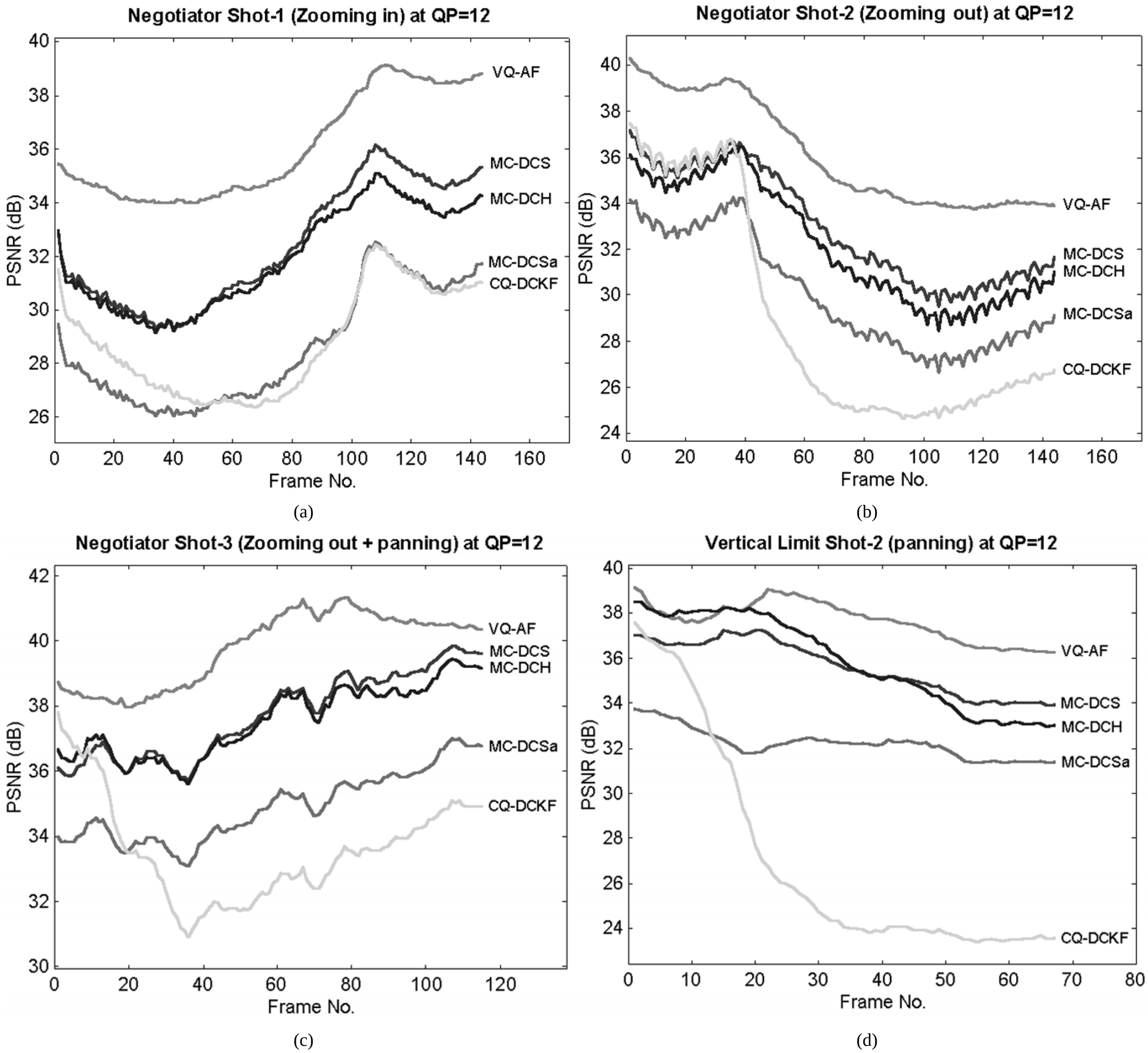


Fig. 6. PSNR performance of various palette generation schemes at QP = 12 under different conditions: (a) zooming in; (b) zooming out; (c) zooming out + panning; and (d) panning.

REFERENCES

[1] P. Heckbert, "Color image quantization for frame buffer display," *Comput. Graph.*, vol. 16, no. 3, pp. 297–307, July 1982.

[2] M. Gervautz and W. Pugathofer, "A simple method for color quantization: Octee quantization," in *Graphics Gems*, A. Glassner, Ed. New York: Academic, 1990, pp. 287–293.

[3] M. T. Orchard and C. A. Bouman, "Color quantization of images," *IEEE Trans. Signal Processing*, vol. 39, pp. 2677–2690, Dec. 1991.

[4] S. Wan, S. Wong, and P. Prusinkiewicz, "An algorithm for multidimensional data clustering," *ACM Trans. Math. Software*, vol. 14, no. 2, pp. 153–162, 1988.

[5] B. L. Yeo and B. Liu, "Rapid scene analysis on compressed video Source," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 533–544, Dec. 1995.

[6] S. C. Pei, C. M. Cheng, and L. F. Ho, "Limited color display for compressed image and video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 913–922, Sept. 2000.

[7] "Coding of moving pictures and associated audio for digitized storage media up to about 1.5 Mbits/sec—Part 2: Video,", ISO/IEC 11 172-2, 1993.

[8] ITU Telecommunication Standardization Sector of ITU, "Video coding for low bit-rate communication," ITU-T Recommendation H.263, Mar. 1996.

[9] Y. Linde, A. Buzo, and R. M. Gary, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.