ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

Limited Color Display for Compressed Video

Ching-Min Cheng, Soo-Chang Pei and Lung-Feng Ho

Department of Electrical Engineering, National Taiwan University
T aipei, Taiwan, R. O. C. Email:pei@cc.ee.ntu.edu.tw

Abstract

Many display devices now ada ys still allo w a limited number of colors, called color palette, to be display ed simultaneously. Besides, images and videos in most w orld wide web (WWW) databases are in compressed formats. Therefore, it becomes an important issue to retriev e a suitable color palette from compressed domain in order to have fast and faithful color reproduction for these devices. In this paper, the color palette design methods for compressed videos are presen ted. The proposed approaches use the reduced image rather than the whole image for the color palette design to a void the heavy computation in video decompression. Experimental results show that output image qualit yof proposed methods is acceptable to h uman eyes. In addition, empirical results show that the proposed shifting-window scheme can reduce the main problem of displaying quantized image sequences, screen flicker.

I. Introduction

With the prev alence of multimedia and internet, more and more digital images and videos are av ailable for people to access. The digital image or video format is usually quantized with integer from 0 to 255 for eac h of three color components (e.g. red, green, blue). All possible combinations of three of these values gives 256^3 (or 16 million) distinct colors for full-color digital display. How ev er,due to the costs of high speed video RAM, many current PCs and workstations generally have a single 8-bit frame buffer to allow only a limited number of colors, called color palette, to be simultaneously display ed. If an acceptable output image qualit yis desired, it is necessary to develop a useful procedure, called color quantization, for designing the color palette.

In the past, the color palette design focused on uncompressed data. F or one single image, several color quantization algorithms hav e been proposed. Heckbert has proposed a median cut (MC) algorithm[1] where the color space is recursively divided into M rectangular regions with equal color occurrence and their centroids being representation colors. Recently, the popular v ector quantization (V Q) technique[4] is applied to the color palette design, too. The colors in input image are used as training vectors in order to refine the initial rough palette. But VQ-type algorithms are usually computational intensive such that they are not suitable for real-time processing. P ei and Cheng hav e suggested a dependent scalar quantization (DSQ) algorithm[2], which exploits dependency of input colors and sequentially partitions the color space. The experimental results show that the DSQ can reduce the computation complexity and its output image quality is acceptable to human eyes. Also some color quantization researc hers hav e focused on the processing of image sequences. Roytman and Gotsman[8] have proposed an algorithm for dynamic color quantization of image sequences, which quantizes each image independently to produce a different color palette for eac h image. Besides, they suggest the approach of color palette filling to prev ent frequent switc hingof color palettes, which leads to the problem of screen flick er.

How ev er,with the consideration of communication bandwidth and storage space, most image or video data no w ada ys comes in compressed format. Extra hea vy computation of image or video decompression is required before any above mentioned color quantization is employ ed. How to design color palette directly from compressed data has thus become a significant issue. In this paper, w e extend the DSQ to present some nov el color palette design methods for compressed videos. The proposed methods use the reduced image in compressed domain to design color palette. w e propose a shifting-window scheme to display longer sequences without screen In section II, w edescribe the proposed approach for compressed videos, which uses the reduced image rather than the whole image for the color palette design to avoid the heavy computation. In addition, the technique of extracting key frames for compressed videos is presented and a shift-window scheme is proposed to solve the screen flicker problem. Section III reports the simulation results of proposed methods and some discussions are made.

II. Limited Color Display for Compressed Video

Now more and more video clips are able to be accessed in some World Wide Web databases. However, with the limit of bandwidth of in ternet, those video clips are usually compressed b y using MPEG [3] format. In this condition, how to analyze the video clips to design a suitable color palette for those machines with limited color display becomes important. In this section, we will introduce methods of color palette design for MPEG compressed video.

The low-resolution DC images of MPEG video, called DC sequence, are first extracted for color palette design. Additionally, to avoid huge computation costs, w e will only apply the DSQ to those DC images called key frames, which contain the major color information of the entire DC sequence, for obtaining the color palette of MPEG video. F rom theobservation of DC sequences, w enoticed that color information inside a frame is unchanged for most of the sequence except key frames, which consists of color changes in the sequence. Thus, detection of color changes is essential for finding key frames. The detection of k ey frames is based on two steps and introduced in the following subsection:

1. detection of potential key frames
2. detection of k ey frames and extraction of color palette

Besides, if only one fixed color palette is used, the degradation would get worse and worse as the sequence length gets longer. To ov ercome this problem, a multiple color palette scheme called shifting-window scheme is proposed to display compressed video with good quality even when the sequence length is getting longer.

A. Extraction of DC sequence

We will use MPEG-1 video as the example to illustrate the process of DC sequence extraction. In MPEG-1, the DC coefficient of the DCT bloc kin I frame is the av erageof pixels in the bloc k. The DC image for I frame can be easily formed by getting the DC coefficient in every block. How ev er, the c hallenge exists in extracting DC images from P or B frames, which use motion compensation to exploit the temporal redundancy. A general case for P framhas been proposed by Yeo and Liu[7] and is show ed in Figure 1.

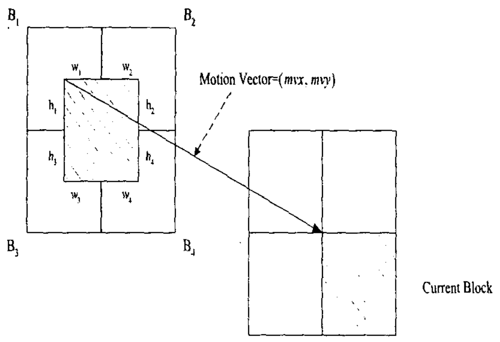


Figure 1: The extraction of DC image.

Here, B_{ref} is the block with motion vector (mvx, mvy) pointing to the current bloc kof in terest and B_1, B_2, B_3 and B_4 are the four neighboring blocks which derive the reference block B_{ref}. We can express the DC component of DCT coefficients of B_{ref} denoted as DCT(B_{ref}) in the following equation.

$$(\mathrm{DCT}(B_{ref}))_{00} = \sum_{i=1}^{4}\sum_{m=0}^{7}\sum_{l=0}^{7} w^i_{ml}(\mathrm{DCT}(B_i))_{ml} \quad (1)$$

where w^i_{ml} are weigh ting factors related to the motion v ector.

This precise calculation of DCT coefficients is time-consuming. Since w e are only interested in the DC component, the first-order approximation is used rather than precise calculation.

$$(\mathrm{DCT}(B_{ref}))_{00} = \sum_{i=1}^{4} \frac{w_i h_i}{64}(\mathrm{DCT}(B_i))_{00} \quad (2)$$

where w_i and h_i are the ov erlapping width and height of B_{ref} in block B_i. It can be sho wnthat $\frac{w_i h_i}{64}$ corresponds to w^i_{00} in Eq. 1 and this is why it's called first-order approximation. Although the approximation error will accumulate, the error is acceptable in most cases because the GOP size is usually small and the error will reset to zero at every I frame. The other advantage of this approximation is that it requires only the motion vector information and the DC values in the reference frames. This approximation approach can also be applied to B frame where tw o reference frames might beneeded. The problem of half-pixel-wise prediction can be solv ed by using the av erageof larger bloc ks depending on the motion vector. The 9x9 block is needed if half-pixel-wise prediction occurs in both x and y direction. The 8x9 bloc kis used for half-pixel accuracy only in x direction and 9x8 block is for y direction only.

B. Key F rame Detection for Compressed Video

After the DC sequence is deriv edfrom a MPEG video, w euse its DC images for key frame detection. T orepresent color information of a frame, hue component of HSV space is adopted in this paper. Hue component has been widely accepted as a good candidate of representing color difference. Using this represen tation, w ecompare the normalized difference of hue histogram betw een consecutive frames in order to detect boundaries betw eenconsecutive color changes. The principle behind this is that tw oframes having a unchanging background and objects will show little difference in their corresponding hue histograms. The normalization procedure is utilized for reducing the impact of noise imposing on the hue histograms of consecutive frames. The normalized hue difference betw een the hue histograms of the lth and $(l-1)$th frame, S_l is given by the following equations:

$$S_l = \sum_{j=1}^{L} NHD_{l,l-1}(j) \quad (3)$$

where

$$NHD_{l,l-1}(j) = \begin{cases} H_l^3(j) & \text{for } H_{l-1}(j) = 0 \\ \left|\frac{H_l(j)-H_{l-1}(j)}{min(H_l(j),H_{l-1}(j))+1}\right| & \text{otherwise} \end{cases} \quad (4)$$

with $H_l(j)$ and $H_{l-1}(j)$ being hue histograms of the tw o consecutive frames respectively and L of Eq. 3 being the number of hue component bins in comparison. In Eq. 4, we choose the minimum value of $H_l(j)$ and $H_{l-1}(j)$ to normalize the hue difference. The condition, $H_{l-1}(j) = 0$, is set to reflect the situation when pixels with the certain hue value j exists in frame l, but not in frame $l-1$.

Similar to luminance change of the DC sequence, we ha ve observed that color c hangeis also a local activity which inv olv es details regarding several neighboring frames. For scene change analysis, Yeo and Liu [5] has suggested to set the threshold of luminance change in order to match the local activit y. We adopt this approach to detect color changes in this paper and choose a sliding window thresholding technique proposed b y Y eo and Liu [5] to av oid false alarms which might occur in camera operations or object changes. In this technique, $2n-1$ frames with $2n-2$ h ue differences are examined in a local range. Inside this local window, a color change from $(l-1)$th to lth image occurs if the following conditions are satisfied:

1. The difference is the maximum with the window, i.e., $S_j \le S_l, j = l-n+1, \cdots, l-1, l+1, \cdots, l+n-1$.
2. S_l is also m times of the second maximum in this window.

After examination of each window, the window is shifted one frame to prepare the next examination until the whole sequence is processed. In criterion 1, the parameter n is set to be smaller than the minimum duration betw een t w o colorhanges, but large enough to av oid false alarms. This is because as the window size gets smaller, the threshold is closer to be a global approach which is unfav orable for color change detection.

If we set $n = 30$ for a 30 frame/sec video sequence, it means that there cannot be tw o color changes within a second. The parameter m in criterion 2 is imposed to guard against some camera operations such as fast panning or zooming. F or these operations, the hue differences S_l w ould maintain consecutive peaks across sev eral frames. F rom experimental results, w e understand that the design of m depends on the tradeoff betw een increasing the detection rate and decreasing the false alarm rate. It has been found that the values of m varies from 2.0 to 4.0 give good results.

Through the above sliding-window thresholding scheme, the detected frames are called potential key frames. F rom experimental results, w e observed that there are redundant false-alarmed frames, which don't contain significant color information, inside these poten tial key frames. Then, w e adopt a coarse-to-fine strategy to eliminate those false-alarmed frames. In this strategy ,these potential key frames is processed one more time by the sliding-window thresholding scheme. After this examination, the detected frames are desired key frames which are then used by the DSQ for the extraction of color palettes. We illustrate the proposed scheme of detecting key frames for compressed video in Figure 2.

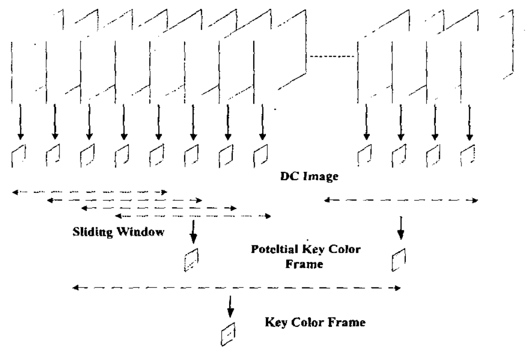


Figure 2: Detection of key frame.

T o do limited-color display with the color palette extracted by the proposed scheme, w e employ the same procedures as for the compressed image. Using the w ell-organized boundaries of color cubes partitioned by the DSQ, pixels of each image of decoded MPEG sequence are mapped to their associated representativ e colors.

C. Shifting-Window Scheme

If only one fixed color palette is assigned to the whole sequence, the performance of color quantization is getting worse as the sequence length gets longer. On the other hand, if a color palette is designed for every k ey frame, a serious visual artifact, screen flicker, may occur when the color palette is changed for these key frames[8]. This problem happens because there is a sudden change of images colors. When the frame buffer of the display con tains an image, a new color palette which belongs to the next image is already active. This phenomenon of screen flicker is sharp and unpleasant to h uman eyes.

To solve the screen flicker problem, we still use the DSQ to design color palettes for the sequence. But a color palette is designed for each fixed-length shifting-window in the sequence. The procedures of color palette design in the video sequence of eac h window are the same as mentioned above. The key frames in eac h shifting-window are detected and applied to the DSQ for the color palette design as depicted in Figure 3. These windows contains o verlapping frames, which causes the color distribution would not vary too muc h from window to window even if the color change occurs. As a result, the DSQ can generate smoothly varying color palettes for the sequence if the bit allocation procedure is fixed. In addition, since every en try of the color palette of the processed window w ould not differ significantly from that of the next window, screen flic ker is greatly reduced.

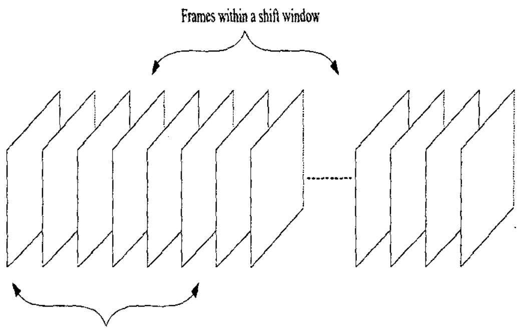


Figure 3: The proposed shifting-window scheme.

III. Experimental Results and Discussions

To evaluate the performance of the proposed method for compressed videos, a test sequence "News" which has 300 YUV frames with size 352x240 is used. The sequence consists of three main video segments which are news conference of a president candidate, a TV news reporter and news of a tropical storm. There are special effects of dissolving, fading in and fading out existed within each transition of tw o segments. This test sequence is first compressed by MPEG-1 compression. Then the proposed scheme is applied to design a color palette of 256 colors on the YUV color space. The color components were quantized in the order of Y(Luminance) first, then U and V last. The numbers of bits in each color component after bit allocation in the DSQ are 4(Y), 2(U) and 2(V).

The detected key frames from the extracted DC sequence are frame 0, 3, 90, 195 and 210 which are shown in Figure 4. The parameters of criterion 1 and 2 used to find a color change are set to be 7 and 2, respectively. The sum of hue difference in the first step of scheme of detecting key frames is plotted in Figure 5. As w e can see, these frames indeed represent significant color difference. F rame 210 is detected because the yellow caption appears. And w e plot in Figure 6 the PSNR distribution of luminance component of the decoded MPEG-1 sequence and the decoded sequence quantized by the proposed scheme. In this figure, the average PSNR of the decoded MPEG-1 sequence is 33.9805 dB and that of the proposed scheme is 32.4647 dB, which shows only about 1.5 dB lost on average in the color quantization. When we analyze Figure 6, it is noticed that some peaks or intra-frames among frame 90 and frame 210 have about 5 dB lost betw een PSNR values of the decoded MPEG-1 sequence and the proposed scheme. Normally, decoded intra-frames of MPEG-1 sequence are good quality since no motion estimation is involv ed. F or these intra-frames, it shows that the matching of colors of decoded MPEG-1 frame with those of the original frame is better than that of representative colors obtained from the proposed

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

Figure 4: The DC images of detected key frames: (a)frame 0 (b)frame 3 (c)frame 90 (d)frame 195 (e)frame 210

scheme. How ever,since the PSNR values of the proposed scheme for these frames are around 35 dB, we don't perceive significant degradation of picture quality in experiments. Concerning the computation time, the proposed method took about 8.8 seconds to extract a color palette for the test sequence when using a SUN SPARC20 workstation.

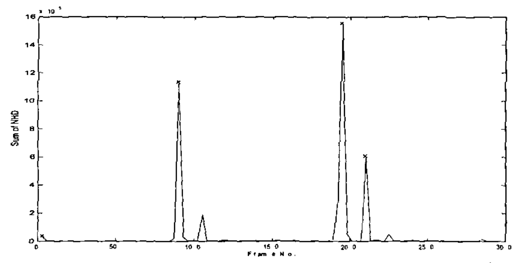


Figure 5: Sum of hue difference plot of "News".

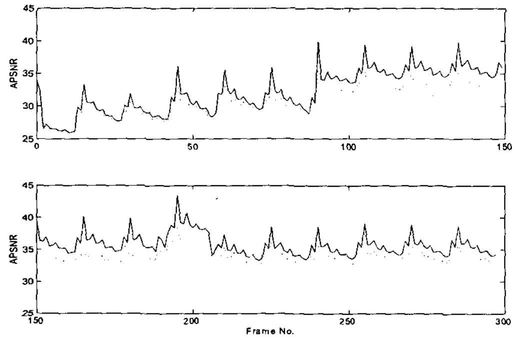
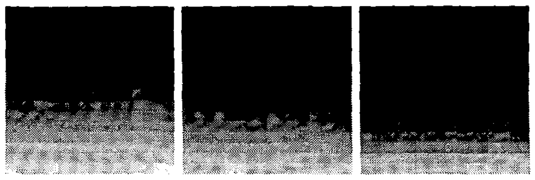


Figure 6: The PSNR in the sequence "News". The solid curve is for the decoded MPEG-1 sequence and the dotted curve corresponds to the proposed scheme.

We also executed the shifting-window scheme with the size of shifting-window being 150 and the size of overlapping frames between neighbouring windows being 75 on the test sequence. This configuration results in three shifting-windows to cover the test sequence. The designed color palettes for frames 0-149, 75-225 and 150-299 are shown in Figure 7(a), (b) and (c). As we can see, the color palette of frames 150-299 contains more dark colors which appear in the video clip of the tropical storm. And the gradual changes can be observed among these color palettes. When the corresponding quantized sequence is played back with frames 0-74 using palette of Figure 15(a), frames 75-224 using palette of Figure 15(b), and frames 225-299 using palette of Figure 15(c), we have seen that screen flicker phenomenon is insipid and acceptable to human eyes.



(a) (b) (c)

Figure 7: The color palettes for the sequence "News" when the shifting-windows scheme is employed.

References

[1] P. Heckbert, "Color image quantization for frame buffer display", *Comput. Graph.*, vol. 16, no. 3, pp. 297-397, Jul. 1982.

[2] S. C. Pei and C. M. Cheng, "Dependent scalar quantization of Color Images", *IEEE Trans. Circuits and Systems for Video Technology,* vol. 5, no. 2, pp. 124-139, Apr.1995.

[3] "MPEG Video Committee Draft," ISO-IEC/JTC1/SC29/WE11/MPEG 90/176, Dec. 1990.

[4] R. Gray, "Vector quantization", *IEEE ASSP Mag.*, vol. 1, pp. 4-29, Apr. 1984.

[5] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Videos", *IEEE Trans. Circuits and Systems for Video Technolgy,* vol. 5, no. 6, pp. 533-544, Dec. 1995.

[6] Y. Nakajima, "a Video Browsing Using Fast Scene Cut Detection for an Efficient Network Video Database Access", *IEICE Trans. Inf. and Syst.*, vol.E77-D, no. 12, pp. 1355-1364, Dec. 1994.

[7] B. L. Yeo and B. Liu, "On the extraction of DC sequences from MPEG compressed video", *International Conference on Image Processing*, vol. 2, pp. 260-263, Oct. 1995.

[8] E. Roytman and C. Gotsman, "Dynamic color quantization of video sequences", *IEEE Trans. Visualization and Computer Graphics*, vol. 1, no. 3, pp. 274-286, Sep. 1995.