ABSTRACT

We developed a very low bit rate watercolor-like video form, which is constructed with a bi-level Y luminance component and a number of UV chrominance combinations. The bi-level Y-component outlines the features of an image and the UV-combinations describe the basic color information of the image. The UV combinations are dynamically chosen during encoding. The resultant video looks better than bi-level video and its bit rate is still much lower than that of full-color video. Watercolor video can be used in video broadcast and communication in very low bandwidth networks.

1. INTRODUCTION

Network services are being developed to empower users to access information anywhere, anytime on any device. Video as an important form of information or media is a powerful enhancement for human communication. While it is relatively easy to transmit popular MPEG [1] or H.26x [2] video through broadband networks, it is not so easy to transmit such kind of video through wireless networks with limited bandwidth. The challenge of transmitting video through any networks is usually in the lower end.

As a solution to transmitting video over low bandwidth networks, we previously proposed a video form, called bi-level video [3]. Bi-level video is generated by converting a gray-scale image to a bi-level image and then compressing a bi-level image sequence into a bi-level video bitstream. Special considerations are taken in the thresholding and compression processes to enable the resultant video with nice properties such as, very low bit rate, sharp shape, smooth motion, and low latency.

Although bi-level video achieves very low bit rate, it only provides black and white pictures and therefore it is not always satisfactory. On the other hand, in MPEG video, the UV components normally only take about 5-10% of the total bit rate. Can we improve the overall visual quality of bi-level video by adding only a few of UV components while keeping the bit rate very low?

The answer is yes. By converting a gray-scale image to a bi-level image, we reduced the complexity of the Y component but preserved the most important outline features if the threshold is selected properly. The addition of a few UV components will provide more color features to the white pixels in the bi-level images. As shown in Fig.1 such kind of video looks like painted using watercolor – a type of paint composed of a water-soluble pigment, which displays basic colors of a picture.

Figure 1: (a) Full-color image, (b) watercolor image and (c) bi-level image.

In section 2, we describe the compression algorithm of watercolor video. The algorithm is first applied to the I-frames and then we discuss how to deal with the P-frames. Experimental results on test video clips are shown in section 3. We conclude this work and point out future directions in section 4.

2. ALGORITHM

The key techniques in watercolor video coding are how to describe the chrominance information of an image and how to select representative chrominance information and compress it.

2.1. What Kind of UV Components Are Needed?

The proposed watercolor video is based on our experience in both bi-level video coding and MPEG video coding. In MPEG video coding, each picture is divided into some blocks. The block size is usually 16x16 for Y component and 8x8 for UV. The resolution of the U and V components is usually half of that of Y. The Y, U, and V values in each block are transformed from the spatial domain into a set of DCT coefficients in the frequency domain. Each of these coefficients is a weight corresponding to a DCT basis waveform. These coefficients are then quantized, and then compressed using an entropy coder. In our proposal, since we only use a bi-level Y component, we don’t need very accurate UV distributions either. Experimental results shown in the latter part of this paper will justify this consideration. Obviously, the most important part of the DCT coefficients of each U and V block is just the first coefficient, i.e. the DC component. The physical meaning of the DC component is the average chrominance of the block.
2.2. How Many Typical UV Combinations Are Needed?

Since watercolor video is specially developed for very low bit rate video communication, we only need to consider small resolution images. Take a QCIF image for example. If we divide a QCIF (176×144) image into 16x16 blocks, the total number of blocks is 99. So there are totally 99 UV combinations after the above averaging process. However, if we just use these 99 UV combinations, the resultant data will be very diverse. In order to obtain a limited number of representative UV combinations, a VQ-like process is used. We first round the values of U and V to the nearest multiples of 4. The total number of UV combinations is reduced to about 20-30 according to our statistics. We setup totally 16 indices to represent the most frequent 16 UV combinations. For the remaining UV combinations, their 2-dimensional Euclidean distances from each of the 16 representative UV combinations are calculated, and each of the remaining values is merged to the nearest representative UV combination.

2.3. Compressing the UV Components

Now, we have a UV combination lookup table which stores the U and V values of 16 typical UV combinations of an image. Each 16x16 block of an image is assigned with an index of the table. Huffman coding is used to encode these indices. We calculate the probability for each entry in the lookup table in terms of its frequency and construct binary codes from probabilities using Huffman coding. For example, when we compress the indices of blocks in the QCIF Akiyo video sequence, we get an average code length of 3.86 bits, therefore the UV components of each frame occupy only 99×3.86 = 382 bits.

2.4. P-frame Consideration

The above procedure is first applied to the I-frames of a video sequence. For P-frames, the averaged UV components of each block are classified to the typical UV combinations of I-frames. This treatment will not cause obvious errors because in a low bitrate video communication scenario, people are not very sensitive to details of a small scene, and the background is usually not changed. The central part of the scene is usually the user’s face which possesses only limited color types. Since we can use the same lookup table of an I-frame for its subsequent P-frames, we need only transmit the lookup table which stores the U and V values of the typical UV combinations of each I-frame.

2.5. Decoding and Displaying

The bit stream of each frame in a video sequence is composed of a bit stream of a bi-level Y component and a bit stream of UV components. In the decoder, the bit stream of the bi-level Y component and the bit stream of the UV components are decompressed and output to display. As we have mentioned above, block artifacts usually occur since we only use the average UV values of a 16x16 block. On the display side, we use bilinear interpolation to eliminate them. For pixels that are located along the edge of the image, their UV values are just assigned with the UV values of the block.

2.6. Optimizing the Selection of Typical UV Combinations

As we have described above, the lookup table is composed of 16 combinations with the highest statistical probabilities. If the chrominance of an image is distributed relatively uniformly, i.e. the differences of the probabilities of these UV combinations are relatively small, this method will lead to evident errors. In order to overcome this drawback, we adopt Lloyd-Max Quantization theory [4] to obtain the optimal UV combinations through iterations. The procedure is as follows:

(1) Reassign the U and V values of each UV combination to the nearest multiple of 4.
(2) Select the 16 most frequent UV combinations.
(3) All the other UV combinations are merged to the closest entry in the current set of 16 UV combinations according to their 2-dimensional Euclidean distances.
(4) Calculate the average of the U and V values of each entry and reassign the results to the entry to compose the new 16 UV combinations.
(5) If the U and V values of the 16 UV combinations no longer change, or the iteration reaches a maximal number, the iterations stop; otherwise go to step (3).

2.7. Further Optimizing the UV Combinations

In the display side, the bi-level Y component and UV components are combined to produce the final color. For those UV combinations that are generated from blocks in which the Y component value is 0, the final color in the display is almost black. In other words, they do not evidently affect the appearance of the watercolor image. So UV combinations that are only generated from Y=0 blocks could be ignored in the statistics collecting process. This results in more indices being assigned to UV combinations that visually affect the watercolor image. In practice, it is hard to find a block in an image where all the Y component values are 0, so we disregard the UV combination of a block if over 90% of the Y component values are 0. As described in subsection 2.4, the scene in P-frames is very similar to that of the I-frame, therefore, this method will not obviously affect the quality of P-frames. Experiments on the QCIF Akiyo video clip show that the final average code length of each block is reduced to 2.28 bits with this method.
3. EXPERIMENTAL RESULTS

We tested our approach on both standard MPEG test video clips (Fig.2 (a), (b), (c)) and ordinary clips captured from real scenes using PC digital cameras (Fig.2 (d)). Video clip (a) Akiyo is representative of scenes with little head motion, clip (b) Grandma is a typical scene with relatively large head motion and clip (c) Silent possesses large motion of arms and hands. Video clip (d) Yu represents scenes with very large motion of the whole body. All the video clips are in QCIF format at a frame rate of 30 fps. We compared the visual effects and frame rates obtained by our method with one of the H.263+ implementations [5].

Fig.2 (a) through (d) show one typical frame of each clip coded with the H.263+ encoder, the watercolor video encoder and the bi-level video encoder respectively. It is shown that the images (with subscript 1) coded with the H.263+ encoder have some blocks which result in the artifacts on Akiyo, Grandma and Yu’s faces and Silent’s whole body, while the images (with subscript 2) coded by the watercolor video encoder are as sharp as the images (with subscript 3) coded by the bi-level video encoder. Furthermore, the basic colors of the original clips are kept and the whole visual effect is obviously better than that of the bi-level images.

The smoothness of motion is a feature of watercolor video. Fig.3 (a) through (d) show the frame rates of each clip coded by the watercolor video encoder and the H.263+ encoder. The frame rates of coded video clips are measured when bandwidth is set to 4.8, 7.2, 9.6, 14.4, 19.2 and 28.8 Kbps, and the I-frame interval is set to 5 seconds. Experiments show that in most cases the watercolor video encoder generates higher frame rates than the H.263+ encoder. Therefore, watercolor video provides smoother motion of scenes. The exceptions are those low motion clips, clip (a) Akiyo and (b) Grandma, where the frame rates generated by H.263+ encoder are a little higher than those produced by the watercolor video encoder when the bandwidths are about 4.8 Kbps. This is because at such bit rate, even if we remove the chrominance information from the watercolor video, the frame rates generated by the two encoders are very close. Note that at such bit rates the H.263+ suffers from severe blocking artifacts while watercolor video doesn’t. On the other hand, given the same conditions in clip (c) Silent and (d) Yu, the large motion made the frame rates produced by H.263+ encoder lower than those obtained by the watercolor video encoder.
4. CONCLUSIONS

We have developed a low bit rate watercolor video form. For an input video, its Y component is compressed using bi-level video coding. The UV components are first averaged in each block, and then represented by a set of typical UV combinations (codebook). These indices are further encoded using Huffman coding. The lookup table is transmitted with each I-frame. For P-frames, the chrominance information of each block is just coded using the lookup table of its corresponding I-frame.

Experiments on standard MPEG test video clips and captured clips of ordinary scenes show that watercolor video looks better than bi-level video, yet its frame rate is higher than that of full-color video at the same bit rate.

Future research directions could include how to establish a scalable coding method of chrominance and how to balance the bit rate allocation in the Y component and the UV component to reach the best quality for any given bandwidth.

5. REFERENCES