# WEIGHTED MOTION ESTIMATION FOR EFFICIENTLY CODING SCENE TRANSITION VIDEO

*You Zhou*[*1], *Xiaoyan Sun*[2], *Hong Bao*[1], *Shipeng Li*[2]

[1] University of Science and Technology Beijing, Beijing 100080
[2]Microsoft Research Asia, Beijing 100080

## ABSTRACT

Scene transition video has brought great challenges to current video coding methods because the traditional motion model of block displacement can not efficiently represent transition motion. This paper first analyzes the features of static transitions. By incorporating the information of transition filter into the coding scheme, the weighted motion estimation (WTME) technique is proposed to get accurate motion parameters of transition video, thereby efficiently compensating both normal and transition motions among frames. Experimental results show that the proposed technique can significantly improve the coding performance of H.264 up to 2.0dB while coding scene transition video.

## 1. INTRODUCTION

Nowadays, many video contents, especially news, movie and music TV, often contain scene changes, which are generated by processing raw video with transition filters. Moreover, together with the rapidly increasing in the amount of the digital camera and camcorder users, more and more people are enjoying to create their personal and family video by connecting some photos and video clips together with various transition effects. However, since the motion of transition effects is quite different from that of normal camera and objects, it is very difficult for conventional video coding methods to achieve an approving coding efficiency on those transition frames.

Normally, scene transitions are categorized into two types: static scene transition and dynamic scene transition. In the static scene transition, raw video only undergoes pixel amplitude changes but still remains the original orientation and size. That is, no additional motion is introduced by static scene transition. Typical static scene transitions are cross-fade, wipe, checkerboard, circle, and so on. On the other hand, dynamic scene transition introduces new motion upon raw video, where its orientation and/or size may be different from the original (such as moving, rotation and page curling). Due to the complicated motion involved in dynamic scene transitions, they are not discussed in this paper.

Since scene transitions are created by video editing software, it is reasonable to define some transition patterns in common use so that the transition filters are available and can be utilized in both encoding and decoding. Thus the coding efficiency of the transition video can be improved by making use of the transition filters to exactly track the changes caused by transitions. Based on this assumption, some schemes were proposed for coding faded scene transition [1]~[3]. The pixel-amplitude-weighting (PAW) overlay coding scheme proposed by Dong Tian [3] is one of these schemes in which the frames in one scene are coded independently from those in the other scene. In each scene coding, all pixels in the reference frame are first scaled by the faded filter before motion estimation so that the amplitude of the reference is at the same level as that of the current frame. The reconstructed non-scaled frames from the two scenes, called component images, are stored in the buffer and will be integrated by the cross-faded transition for display.

By separately coding two transitional scenes and making use of the faded factors to scale the references before motion estimation, the PAW scheme significantly reduces bits to code the faded transition video compared with common coding methods. But there are several potential problems in PAW that prevent it from further being applied. A major limitation of the scheme is that it is only applicable to faded scene transitions in which the transition factors are changed at frame level. For many other transition patterns, such as wipe, the changes of the transition factors may vary both spatially and temporally. In other words, how to adjust the reference picture should depend on the motion of different regions. Thus, the PAW can hardly guarantee that the pixel amplitudes of the reference are leveled with those of the current picture by only weighting the reference before motion estimation.

In this paper, an efficient weighted motion estimation method is proposed for coding static scene transitions. Similar to PAW, the pictures in two transitional scenes are coded independently in our scheme. However, the motion estimation is performed with regard to the different pixel amplitudes rather than merely adjusting the brightness of the reference pictures, thus the proposed scheme manages to make every pixel in the candidate matching block be in the same amplitude as that in the current block despite of the transition pattern. Moreover, in the reconstructed ref-

[*] This work has been done while the author is with Microsoft Research Asia.

erence, the regions that ought to be black during transitions are filled with appropriate content to eliminate the black-filled artifacts. Experimental results show that the proposed scheme can improve the coding efficiency of static scene transitions up to 2.0dB compared with the normal H.264 coding method.

The paper is organized as follows. In section 2, the weighted motion estimation and compensation are presented. The experimental results are given in Section 3. Finally, Section 4 concludes this paper.

## 2. WEIGHTED MOTION ESTIMAION

During static scene transitions, there are various ways for the second scene to uncover from the first one. In general, they can be formulated as follows:

$$P_f(i,x,y) = F(i,x,y) \times P_\alpha(i,x,y) + (1 - F(i,x,y)) \times P_\beta(i,x,y) \quad (1)$$

where $i$ denotes the picture index, $P_\alpha$ stands for the original pictures in the first scene, $P_\beta$ stands for the original pictures in the second scene, $P_f$ denotes the pictures in the scene transition, $x$ and $y$ denote the position of a pixel, $F(i,x,y)$ stands for the filter function performed on the pixel located at $x$ and $y$ in frame $i$ and should fall within $[0,1]$. As shown in formula (1), during static transitions, all video objects will keep their positions in the original picture without aliasing and additional motion introduced.

**Figure 1 Illustration of wipe transition (from Carphone to Foreman)**

Figure 1 illustrates a kind of static scene transitions, wiping from sequence Carphone to Foreman. In this case, the first scene Carphone is gradually wiped from the second scene Foreman. As shown in Figure 1, two scenes are semi-transparently presented around the central area in the middle frame and the transparency of two scenes is various from the left to the right. Obviously, the transitional video contents are quite different from the normally recorded video sequences. They have to face the special problem that the pixel amplitude varies not only temporally but also spatially. Thus, directly performing motion estimation between the transitional frames may result in affected estimating because the candidate block and the corresponding target block are compared in different pixel amplitudes. As sketched in Figure 2, there is an object (a car) moving from the left to the right during the wipe transition. Due to the amplitude variance between two matching blocks, normal motion estimation may find a "false" block to match the target one, thus drops the coding efficiency of scenes transitions. Similarly, as shown in Figure 3, during the checkerboard transition, the conventional coding method can hardly extract the true motion information in case of the checkerboard artifacts.

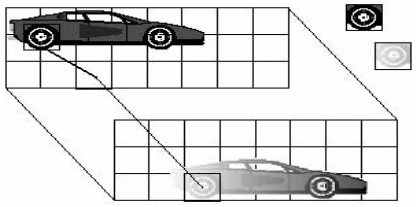

**Figure 2 A car moves from the left to the right during wipe transition**

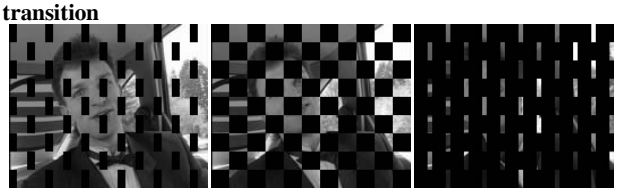

**Figure 3 Illustration of checkerboard transition (Carphone)**

By taking advantages of the available transition filters, the weighted motion estimation (WTME) method is presented in this paper to provide high coding performance of the transition frames. For easily analyzing the transition process, as well as in PAW, the scene transitions are decomposed into to-black process and from-black process in the proposed WTME scheme. Also, the two sequences in transition are investigated separately and correspondingly named as to-black subsequence and from-black subsequence. As instanced by Figure 1 and Figure 4, though the wipe transition is performed on both Carphone and Foreman, the Carphone sequence is wiped to black and encoded independently from the from-black Foreman sequence. At the decoder, the two subsequences are decoded separately and integrated with the wipe effect for display.

**Figure 4 To-black subsequence of the wipe transition illustrated in Figure 1 (Carphone)**

Similar as in the PAW scheme, the from-black subsequence is coded in reversed temporal order in our scheme since it is beneficial to predict a picture containing less content from a picture containing more. Therefore, the problem handled in the proposed WTME is simplified to how to encode the to-black subsequence efficiently.

As we know that the purpose of motion estimation is to remove temporal data redundancy and therefore, attain high compression ratios [4]. Conventionally, the problem for motion estimation to solve is to determine a matching block in reference frame so that the cost of representing the motion vector and prediction error measured as *SSD* or *SAD* is minimized. The *SAD* error can be formulated as follows [5]:

$$E((\Delta x,\Delta y)) = \sum_{(x,y)\in\Lambda} \left| P(x,y) - \tilde{P}(x-\Delta x, y-\Delta y) \right| \quad (2)$$

where $P$ stands for the current frame, $\tilde{P}$ denotes the reconstructed reference stored in frame buffer. $\Lambda$ is the domain of all pixels in the current block and $(\Delta x,\Delta y)$ is the candidate motion vector. However, since the motion estimation is performed between two frames with different pixel amplitudes in transitions, normal motion estimation methods can not adequately eliminate the temporal redundancy between frames and will drop the coding efficiency of scene transitions.

Moreover, unlike natural sequences, to-black sequences contain less and less real video content along with the transition's going on, it is reasonable to spend fewer and fewer bits coding the to-black transitional frames. However, as shown in Figure 3, more and more block areas, together with the sharp edges between the black areas and the non-black areas introduced by scene transitions, will seriously spoil the dependency inherently existing between two continuous frames.

In order to reduce the influence of transitions on motion estimation, in the proposed method, the value of each pixel in $\tilde{P}$ is scaled by a weighted factor $f$ during motion estimation. Suppose that

$$\begin{aligned} f_{org} &= F(i,x,y) \\ f_{ref} &= F(i-\Delta i, x-\Delta x, y-\Delta y) \end{aligned} \quad (3)$$

where $\Delta i$ stands for the temporal distance between the current frame and the reference. $F$ stands for the filter function. $f_{org}$ and $f_{ref}$ denote the transition factors of every comparing pixels in the current block and the candidate block, respectively. Thus, the weighted factor $f$ of each pixel is defined as

$$f = \begin{cases} f_{org} / f_{ref}, & f_{ref} \neq 0 \text{ and } f_{org} \neq 0 \\ 1, & otherwise \end{cases} \quad (4)$$

In fact, if the transition factor of the original pixel is zero ($f_{org} = 0$), nothing needs to be coded since the transition filter will "tell" the decoder that it is zero. While when only $f_{ref}$ is zero, the motion estimation is process on the non-scaled pixels as shown in formula (2).

Utilizing the weighted factor $f$, the $SAD$ in the proposed WTME method is accumulated as follows:

$$E((\Delta x,\Delta y)) = \sum_{(x,y)\in\Lambda} |S|, \quad (5)$$

where

$$S = \begin{cases} P(x,y) - f \times \tilde{P}(x-\Delta x, y-\Delta y), & f_{org} \neq 0 \\ 0, & f_{org} = 0 \end{cases}. \quad (6)$$

Figure 5 illustrates the proposed WTME encoder in which the motion estimation is performed between original frames stored in $F_n$ and the reference stored in $F_{ref}$. As shown in Figure 5, different from a normal encoder, the transition filter is involved into the motion estimation to scale every pixel with the weighted factor $f$. That is, when searching the matching block in the reference, every pixel in the candidate block is weighted in the proposed WTME method. Thus for the transitions in which the factors are changed at pixel level, such as wipe, WTME is able to make all the pixels in the candidate block always in the same amplitude as those in the current block in motion estimation. As a result, the coding efficiency is significantly improved for coding such scene transitions.

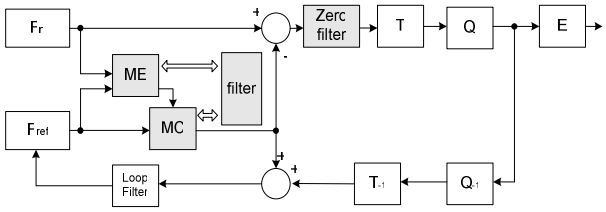


**Figure 5 The diagram of the proposed WTME encoder**

Correspondingly, as shown in Figure 5, the transition filtering also plays an important role in motion compensation, which can be formulated as follows:

$$P_{MCP}(x,y) = f \times \tilde{P}(x-\Delta x, y-\Delta y). \quad (7)$$

Then, the difference between the original frame and the reconstructed scaled reconstructed prediction image forms the prediction errors.

Notice that another distinctive module of the proposed WTME encoder is the "zero filter" by which the prediction errors are forced to be zero if their corresponding pixels are with zero-transition-factors ($f_{org} = 0$). However, the prediction blocks instead of the black blocks are still written back into the reconstructed reference for successive motion estimation. These treatments bring some advantages to WTME. Firstly, large number of bits can be saved when the prediction errors are coded as zero for the pixels with zero-amplitude. Secondly, due to the increasing of the number of the non-black blocks in the reference frame, more information can be used in the next frame motion estimation.

## 3. EXPERIMENTAL RESULTS

Simulations are performed to compare the proposed weighted motion estimation with conventional techniques under JVT software JM-6.1e [6]. Four video coding methods, the proposed weight motion estimation scheme (WTME), the technique proposed in [3] (Simple Overlay), the normal JVT H.264 video coding scheme (H.264) and the PAW method [3], are evaluated in the experiments. Four transition patterns, linear cross-fade, wide width wipe, checkerboard mask and circle wipe, are employed in the experiments. Carphone and Foreman in QCIF are selected as the first and the second scene, respectively. All scene transitions begin at the 2nd picture of Carphone and totally

30 pictures are used to complete the transitions. Table 1 summarizes the testing conditions.

Figure 6 shows the curves of average PSNR versus bit rates in the four static scene transitions. As illustrated in Figure 6 a), when coding the faded scenes, the WTME can provide similar coding efficiency as PAW, while achieve 0.9dB coding efficiency gain over H.264 and simple overlay coding at high bit-rate.

For coding the checkerboard scene transition, as shown in Figure 6 b), WTME achieves coding efficiency gain over 2dB compared with H.264 and simple overlay coding. Here, the sub-block size of checkerboard is set to 16×16.

Table 1 The experiment conditions

| | |
|---|---|
| R-D optimization | not used |
| MV search range | ±16 |
| Frame pattern | IPPPP |
| Frame Rate | 15Hz |
| Motion Compensation | up to 1/4 pel accuracy |
| QP | 20, 25, 28, 31, 35 |
| Scene transitions begin at | 2nd picture |
| Transition span | 30 pictures |

The performance of the WTME when coding wide width wipe scene transition is evaluated in Figure 6 c). In this transition pattern, the first scene is gradually wiped from the second scene from left to right as shown in Figure 1. In a wide width belt (about 80 pixel), two scene are presented simultaneously with different transparency. In this case, the coding efficiency gain of the WTME method is about 1dB compared with H.264.

Finally, the performance of WTME on circle wipe transition pattern is shown in Figure 6 d). In this pattern, with an enlarging circle, the first scene is gradually wiped from the second scene. WTME achieve about 0.7dB coding efficiency gain compared with H.264.

## 4. CONCLUSIONS

This paper proposes a weighted motion estimation technique for coding static scene transitions. By performing motion estimation with regards to the different amplitudes of pixels in transitional pictures, WTME is able to make all the pixels in the candidate block always in the same amplitude as those in the current block in motion estimation. Thus, it can code various static scene transitions efficiently. Simulation results show that more than 2dB coding efficiency gain can be achieved.

However, how to code pictures in dynamic scene transitions still needs further studies.

## 5. REFERENCES


[1] Karl Lillevold, "Improved direct mode for B pictures in TML", ITU-T Video Coding Experts Groups (Question 15), document Q15-K44, August 2000.

[2] Yoshihiro Kikuchi, Takeshi Chujoh, "Improved multiple frame motion compensation using frame interpolation", Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, document JVT-B075, Feb 2002.

[3] Dong Tian, Miska M.Hannuksela, Ye-Kui Wang and Moncef Gabbouj, "Coding of faded scene transitions", ICIP2002, New York, USA, Sep. 22-25, 2002.

[4] Christoph Stiller and Jannsz Konrad, "Estimating Motion in Image Sequences", *IEEE Signal Processing Magazine,* July 1999.

[5] Yao Wang, Jörn Ostermann, and Ya-Qin Zhang, *Digital Video Processing and Communications*, Prentice Hall, pp.163-164, 2001.

[6] JVT Reference Software, version JM 6.1e, Mar 2003.


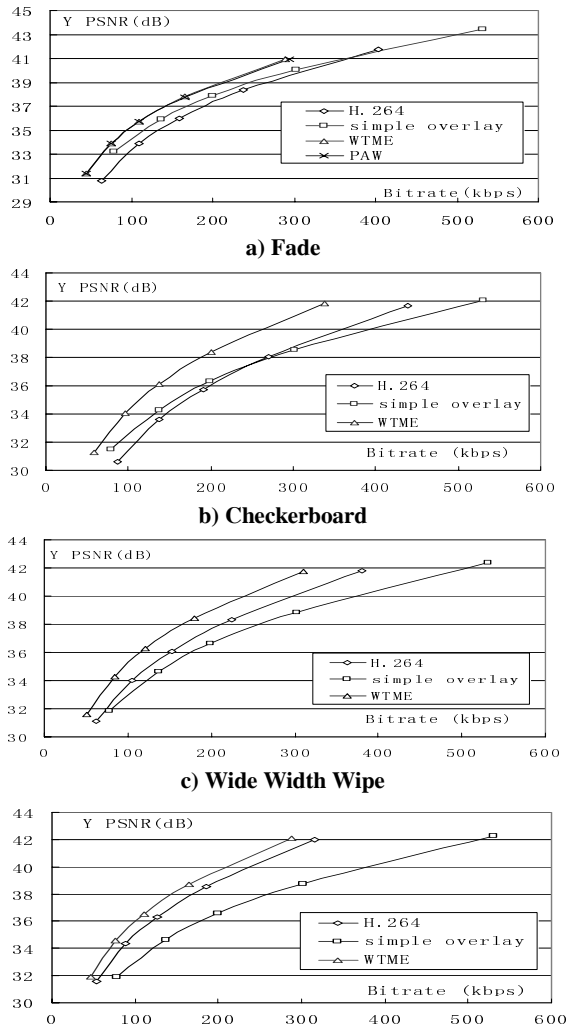


**Figure 6 The curves of average PSNR versus bit rates in different static scene transitions (from Carphone to Foreman)**