New Block-Based Motion Estimation for Sequences with Brightness Variation and Its Application to Static Sprite Generation for Video Compression

Hoi-Kok Cheung, Wan-Chi Siu, Dagan Feng, and Tom Cai

***Abstract*—In this brief, a new local motion estimator is proposed which can accurately estimate motion activities under varying strong brightness conditions. The proposed estimator makes use of a new block division technique which manages practically to get rid of the adverse influence caused by brightness changes between frames. We also propose a new static sprite coding system using the proposed local motion estimator. The system is characterized not only with the features of accurate motion estimation under varying brightness conditions, but also possesses the capability of coding the brightness variability of the background scene using a single layered sprite image. Experimental results show that the resulting static sprite coding system improves the PSNR by 6.32 dB as compared with the conventional static sprite coding system when the background scenes of the video sequences involve strong brightness variations in the spatial and time domains.**

***Index Terms*—Brightness variation compensation, global motions, motion estimation, sprite generation and MPEG-4, video coding.**

I. INTRODUCTION

MOTION estimation is one of the major techniques for video coding. General background motion (global motion) which is primarily induced by the camera motion [1]–[4] can be modeled with a motion model and compactly represented by a set of parameters, called motion parameters. Conventional motion estimators estimate motions based on the brightness constancy assumption [3]–[6]. However, this is not always true as objects may vary in brightness level, depending on the illumination conditions.

In this brief, we propose a new block-based motion estimator which is capable of estimating motion accurately under varying illumination conditions. Our proposed estimator needs no illumination compensation procedure before searching, and the dimension of the search space is effectively kept to two which is identical to other conventional block-based motion estimators. The computational complexity is also comparable. To signify the accuracy of our proposed motion estimator, an algorithm is applied to a static sprite coding system which is relatively sensitive to the accuracy of the local motion estimator. Moreover, our proposed system can effectively handle the variability of the scene illumination in both time and spatial domains.

Manuscript received July 18, 2005; revised April 12, 2007. This work was supported by the National ICT Australia, in part by the Australian Research Council, and in part by the Centre for Signal Processing, The Hong Kong Polytechnic University. This paper was recommended by Associate Editor F. Pereira.

H.-K. Cheung and D. Feng are with the School of Information Technologies, The University of Sydney, Sydney NSW 2006, Australia, and also with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

W.-C. Siu is with the Centre for Multimedia Signal Processing Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: enwcsiu@polyu.edu.hk).

T. Cai is with the School of Information Technologies, The University of Sydney, Sydney NSW 2006, Australia.

Digital Object Identifier 10.1109/TCSVT.2008.918549

II. ILLUMINATION INDEPENDENT LOCAL MOTION ESTIMATION

For an efficient video compression, modeling of the relationship between images is essential in exploring the redundant information. In [7], a generalized brightness change model was proposed

$$I(\mathbf{x}+d(\mathbf{x},t),t+1)=\alpha(\mathbf{x},t)I(\mathbf{x},t)+\beta(\mathbf{x},t) \qquad (1)$$

where $I(\mathbf{x},t)$ indicates the pixel value at location $\mathbf{x}$ and time $t, d(\mathbf{x},t)$ denotes the motion vector and $\alpha\ (\mathbf{x},t)$ and $\beta\ (\mathbf{x},t)$ are called brightness parameters. Barrow and Tenenbaum [8] proposed a simpler multiplicative model by setting $\beta\ (\mathbf{x},t)$ to zero. Generally, these models are too general to be useful and further constraints have to be imposed.

Considering the fact that the brightness variation is slow varying in spatial and time domains, we assume that the brightness parameters are locally constant. Without computing the values of the brightness parameters, we propose in this brief to compute the ratio of the two matching blocks, i.e.,

$$r(\mathbf{x},d'(\mathbf{x},t))=I(\mathbf{x}+d'(\mathbf{x},t),t+1)/I(\mathbf{x},t) \qquad (2)$$

where $d'(\mathbf{x},t)$ is a candidate motion vector. Division by zero is avoided by mapping all pixels with zero value to 1. To measure the quality of matching, we compute the variance of the ratio to check the spatial constancy. The higher the constancy, the better is the matching. The candidate motion vector $d'(\mathbf{x},t)$ giving the minimum variance, which means the highest constancy, is chosen to be the motion vector of the block.

$$d(\mathbf{x},t)=\arg\left[\min_{d'(\mathbf{x},t)\in\mathrm{SW}}[\mathrm{Var}_{\mathbf{x}\in B}(r(\mathbf{x},d'(\mathbf{x},t)))]\right] \qquad (3)$$

where B is the block, SW is the search window, and Var is the variance operator. Let us entitle it as illumination independent local motion estimator (IILME). As the average value of the ratio can vary greatly whenever there is a significant change of brightness, the variance of the ratio is computed to measure the degree of constancy which is independent of the average value of the ratio. More importantly, this merit function has a property that the closer the candidate motion vector is to the true motion, the smaller the variance value. In the application of sprite coding, the performance of coding is sensitive to the accuracy of estimated local motions. Therefore, we deploy a full search scheme to find the best match for a better coding performance. Note also that the computational requirement of the estimator is low as the search space of the proposed estimator is restricted to the search window of the motion estimator which does not include the computation of the brightness parameters.

To visualize the underlying principle of our proposed estimator, let us illustrate the situation of the matching process using a 1-D linear signal. Suppose two identical signals y_0 and y_1 are to be matched. Signal y_1 is transformed using the generalized brightness change model in (1) to simulate the situation that the same signal is observed under different illumination conditions, i.e., $y_0 = mx + c$ and $y_1(d') = \alpha y_0(d)$ where m and c are constants. The relative spatial displacement is set to zero and tested if the proposed motion estimator can estimate the true displacement. In the first case, we consider the multiplicative brightness model, i.e., $\beta = 0$, to model the variation due to illumination change. During the matching process, signal y_1 is shifted by a candidate displacement d' and the quality of the match is computed

$$\frac{y_1(d')}{y_0} = \frac{\alpha[m(x+d')+c]}{mx+c} = \alpha + \frac{\alpha m d'}{mx+c}. \tag{4}$$

The variance of $y_1(d')/y_0$ approaches zero as d' approaches zero from positive or negative sides. Thus, searching for the minimum of the variance $y_1(d')/y_0$ can locate the match position. Taking the computational efficiency into account, instead of computing the division in (2), we compute $\ln(I(x+d'(x,t),t+1) - \ln(I(x,t))$. The log space conversion can be done by using simple table look-up operations prior to the matching. To further reduce the computational complexity, instead of computing the variance in (3), we further propose to compute the sum of the absolute mean difference (SAMD) defined as follows:

$$\text{SAMD} = \sum |r(\mathbf{x}, d'(\mathbf{x},t)) - E(r)| \tag{5}$$

where $E(r)$ is the expected value of $r(\mathbf{x}, \mathbf{d}'(\mathbf{x},t))$. We entitle this motion estimator as BlockDivision(Log).

Considering the more general brightness variation, we have to take into account the offset factor β. i.e., $y_1(d') = \alpha y_0(d) + \beta$. The value of d' corresponding to the minimum of the ratio variance is not zero, which is not the true motion. To remove the biasing factor β, signals y_1 and y_0 are transformed into the mean difference form using their mean values, $E(y_1(d))$ and $E(y_0)$. Thus, the ratio becomes

$$\begin{aligned}\frac{y_1(d') - E(y_1)}{y_0 - E(y_0)} &= \frac{\alpha y_0(d') + \beta - [\alpha E(y_0) + \beta]}{y_0 - E(y_0)} \\ &= \alpha + \frac{\alpha m d'}{mx + c - E(y_0)}.\end{aligned} \tag{6}$$

The variance of the ratio has a minimum when $d' = 0$ which is the true motion. For practical implementation, we cannot compute the ratio in the log space as the shifting procedure creates some negative numbers. Another problem is the computation of the expected values. We can compute the global mean values by taking the whole signal sequence into account, or locally compute the mean values by considering only the pixels within the block. Let us label these two variant local estimators as BlockDivision(GMD) and BlockDivision(LMD), respectively, where GMD and LMD stand for global and local mean difference, respectively. We have tested these three local motion estimators on sprite coding applications. Experimental results show that the coding performance, in terms of coding quality and computational efficiency, for the system using our proposed BlockDivision(Log) outperforms the other two systems. The major reason is that BlockDivion(GMD) and BlockDivision(LMD) depend upon the accuracy of the computed expected values while they are not accurate enough to cancel the effect of β contributing to error.

III. Proposed New Static Sprite Generation

With the static sprite coding scheme, the static nature of the sprite image does not allow a coherent representation of the variability of the lighting condition in different frames. In this brief, we propose a new static sprite coding system for our proposed local motion estimator, which gives an accurate motion estimation under varying illumination conditions. The system also can effectively handle pixel differences caused by brightness change in the spatial and time domains during the sprite construction and frame reconstruction stages. Our proposed system considers only the background objects of the sequence using a segmentation mask. Fig. 1(a) shows an overview of our proposed sprite generation system which mainly consists of four parts, global motion estimation (stage i), global motion refinement (stage ii), illumination parameters estimation (stage iii) and sprite construction (stage iv). In stage (i), we employ our earlier work [9], (which is not repeated here), while the motion estimator uses our proposed IILME instead. It selects a particular image in the past frames to be the reference frame for error accumulation alleviation, and the output of stage (i) is a set of global motion parameters denoting the global motions. Subsequently, the dimension of the sprite is computed and the estimated motions are regulated to compute the long-term global motion parameters. Fig. 1(b) shows the details of the global motion estimation block. In stage (ii), the estimated long-term global motions are refined using our proposed IILME and only the full resolution images are used. Fig. 1(c) shows the details of this stage and the light compensation parameters [α and β in (1)] are estimated using histogram specification techniques.

After the global motion is estimated, we need to transform all the frames to a common lighting condition before constructing the sprite. To fulfill this, we need to code the difference in frame contents between the registered frames due to variations in brightness [stage (iii)]. To express the brightness changes, unlike residue coding, a set of N parameters called illumination compensation coefficients (ICC) are computed for each frame, not for each macroblock, where N is a user defined number determined according to the frequency contents of the illumination functions (see next paragraph for details). Finally, in stage (iv), all frames are motion and brightness compensated to form the sprite image and appear to be under the same lighting conditions. In summary, each frame in the sequence is associated with one set of motion parameters denoting the long-term global motions and one set of illumination compensation coefficients (ICC) describing the brightness variations.

In stage (iii), variations in frame contents due to brightness variations can be coded in two modes. One of them is to use (1) and the histogram specification techniques.

$$\text{hist}(x, \text{current frame}) = \alpha \times \text{hist}(x, \text{sprite}) + \beta. \tag{7}$$

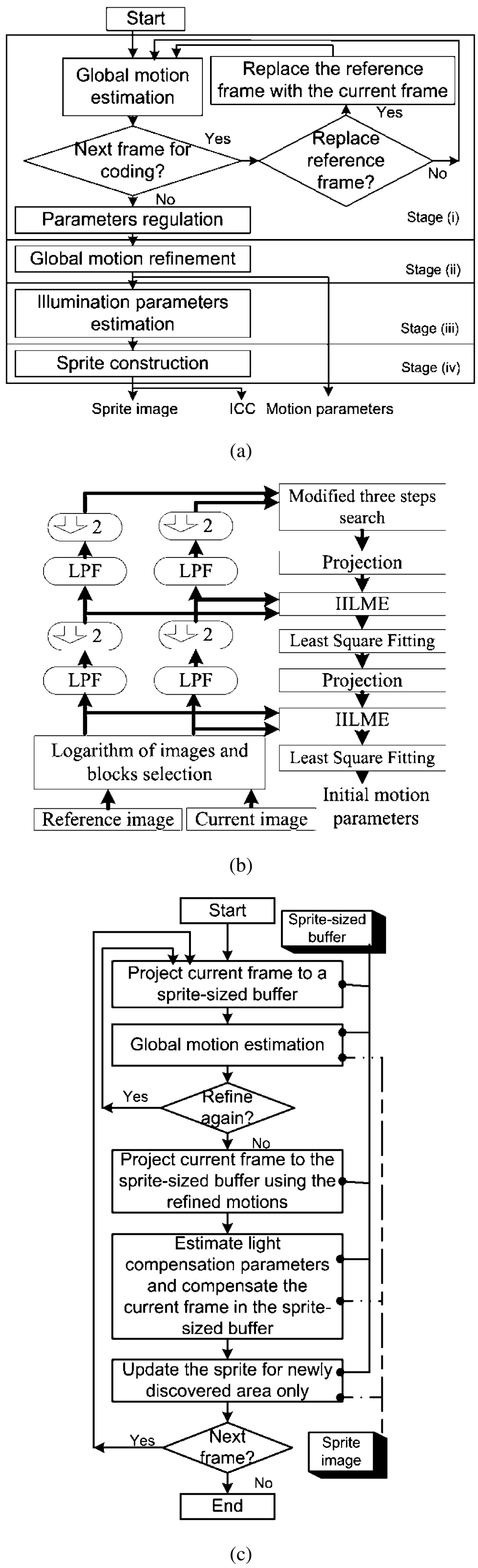

Fig. 1. (a) Overview of the proposed sprite generation system. (b) Block diagram of global motion estimation. (c) Block diagram of global motion refinement procedures.

These two parameters (α and β) represent the ICC using the brightness model-based method, and sprite construction in stage (iv) also make use of these for brightness compensation. This method can effectively explore the correlation between frames due to global temporal illumination variations, which is spatially uniform across frame. In cases where illumination nonuniformity exists, we propose to use the second mode called discrete cosine transform (DCT)-based coding mode.

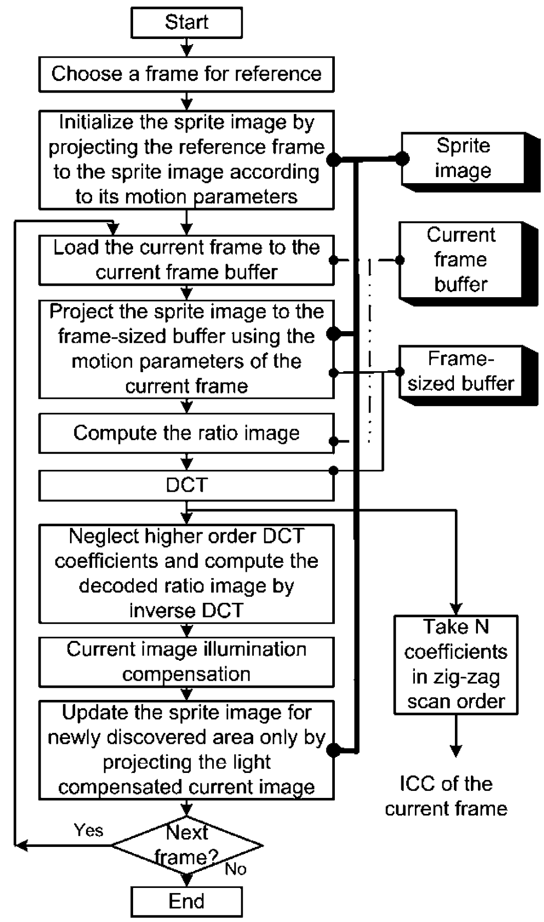

Fig. 2. Block diagram for illumination parameters estimation procedure of the DCT-based method.

The proposed DCT-based method makes use of the multiplicative brightness model and is capable of coding variations due to brightness changes in the spatial and time domains. Fig. 2 shows an overview of the estimation procedure. The ratio image and the current frame illumination compensation are computed using the following equations.

Ratio image

$$= \text{current frame/warped sprite (frame-sized buffer)}$$

Illumination compensated curren frame

$$= \text{current frame/decoded ratio image.}$$

The ratio image conveys the difference between the current frame and the motion compensated sprite. As the difference is constituted mainly by the temporal and spatial brightness variations, and consists of spatially low frequency signals, we encode the ratio image by transforming the whole ratio image using DCT techniques and take the first N low-frequency DCT coefficients in zig-zag scan order. Therefore, using the decoded ratio image, nonuniform brightness variations in the current frame can be compensated for transforming to the lighting condition of the sprite.

To encode the brightness changes, we need to choose a frame for brightness compensation reference purpose. Three choices are considered, the darkest frame, the brightest frame and the frame having the most common brightness level in the sequence. The optimal choice depends on the contents of the sequence and cannot be known beforehand. A suggested method is to attempt all three choices and choose the optimal one in terms of the coding quality. Finally, in stage (iv), all the frames are motion and illumination compensated accordingly and blended together to form the sprite image using temporal average strategy. Similarly, in the decoding stage, frames are warped from the sprite and restore the original brightness accordingly. Note that the conventional residue coding is not used in the coding system.

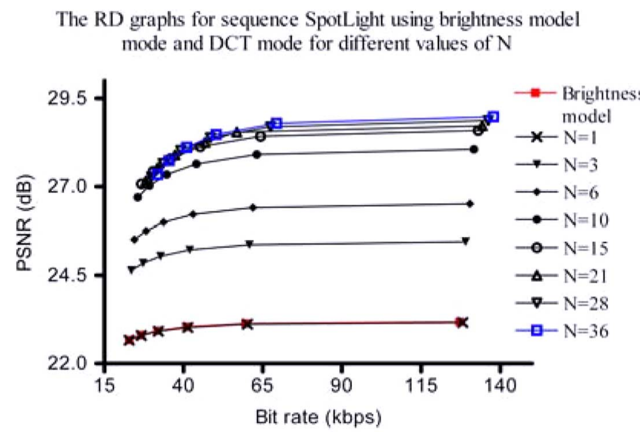

Fig. 3. Rate-distortion diagram for the proposed coder BlockDivision(Log) using the brightness model mode and the DCT mode with different values of N.

IV. Experimental Results

A. Influence of N for DCT Mode on Coding Quality

To study the relationship between the coding performance and the number of DCT coefficients N four test sequences with different properties were used, namely "Stefan" (352×240), "Stefan Alternating" (352×240), "Brighter" (352×288) and "SpotLight" (352×240). "Stefan" and "Stefan Alternating" are provided with segmentation mask and "Stefan" generally does not involve brightness variation. "Stefan Alternating" is originally from "Stefan" in which the even numbered images are downscaled (pixels) by a factor of 1.5 to create rapid, strong and spatially uniform brightness variations. "Brighter" and "SpotLight" involve no global motion. In "Brighter," the only light source is a fixed lamp with a continuously increasing light intensity to generate a nearly spatially uniform illumination condition. In "SpotLight," a moving light source was present which further illuminated a particular area in the scene creating brightness variations in spatial and in time domains.

We ran a series of experiments with the four sequences using MPEG-4 VM, v.18.0. All the results were generated using our proposed system "BlockDivision(Log)" with 100 frames. Fig. 3 shows the rate distortion diagram for "SpotLight," using the two modes with different values of N. The coding efficiency for the brightness model-based system is very close to the DCT-based system with $N = 1$. As N increases, the coding efficiency keeps increasing. An improvement of 6 dB in PSNR has been resulted for N increases from 1 to 36. This significant improvement is within expectation, as significant brightness variations in space and time domain exist. This also implies that using more DCT coefficients to code the brightness difference can solve the deficiency of the multiplicative brightness model and outperform the brightness model-based system.

With "Brighter," a slight improvement of 0.5 dB was found, as the brightness variation in space is not exactly uniform. For "Stefan" and "StefanAlternating," there is no brightness change in space while "StefanAlternating" involves brightness variation in time. Theoretically, the coding qualities are identical for cases using different values of N as only the dc component is important. Experimental results show that the coding efficiencies are very similar to each other, decreasing slightly as N increases. This can be explained by an increase in overheads for coding the extra DCT coefficients. For the rest of the experiments, we arbitrarily chose $N = 10$, based on the result in Fig. 3, as an example to demonstrate the performance of the DCT-based coding mode.

B. Coding Performance Analysis

Table I shows a summary of the coding performance using sprite coders with different motion estimators for the DCT-based ($N = 10$) and brightness-model-based coding modes [each box carries (DCT-based datum/brightness model-based datum)]. The "Traditional PDS" system makes use of the conventional partial distortion search (PDS) techniques, and the brightness changes are not coded. The "Histogram" system makes use of histogram specification techniques to perform global brightness compensation prior to estimating the global motion using conventional PDS techniques. The "MinErrorVar" system performs motion estimation using techniques of minimizing the variance of block difference, i.e., using the additive brightness model with $\alpha = 1$ in (1). Motion vectors are determined by using (3), while $r(\mathbf{x}, d'(\mathbf{x}, t)) = I(\mathbf{x} + d'(\mathbf{x}, t), t + 1) - I(\mathbf{x}, t)$. The "LBC" system (local brightness compensation) and the "GBCLBC" system (global brightness compensation and local brightness compensation) make use of the motion estimation techniques proposed by Kamikura *et al.*[9] and Kim *et al.*[10], respectively. All the coding systems were configured as—top level: search $\text{range} = \pm 6$, intermediate level: search $\text{range} = \pm 2$, base level: search $\text{range} = \pm 4$, refinement stage: search $\text{range} = \pm 4$.

In the table, each group of rows represents the coding results of one test sequence. The first row of each group shows the name (first column) of the test sequence, and this is followed by the number of frames used and the coding quality in PSNR (on the right). The second row shows the chosen reference frame for ICC coding resulting from the best coding performance. The

TABLE I
SUMMARY OF THE CODING PERFORMANCE FOR DIFFERENT CODERS USING THE DCT BASED AND BRIGHTNESS MODEL BASED CODING MODE. EACH BOX CARRIES DCT-BASED DATUM/BRIGHTNESS MODEL-BASED DATUM. FOR MODE ENTRY, D=DARKEST FRAME, B=BRIGHTEST FRAME AND M=MOST COMMON BRIGHTNESS LEVEL FRAME

	Traditional PDS	Histogram	GBCLBC	LBC	MinErrorVar	Block Division (Ori)	Block Division (Log)	Block Division (GMD)	Block Division (LMD)
Stefan 300 [dB]	22.88	22.61 / 22.80	22.18 / 21.88	22.53 / 22.45	22.64 / 22.85	22.62 / 23.10	22.67 / 22.59	22.57 / 22.69	21.68 / 22.69
Mode	-	D / B	D / B	D / B	D / B	D / B	D / B	D / B	D / B
ME [s]	328	396 / 394	424 / 428	503 / 505	347 / 350	394 / 394	371 / 375	405 / 399	407 / 398
Efficiency index	**0.0698**	0.0571 / 0.0579	0.0523 / 0.0511	0.0448 / 0.0444	0.0652 / 0.0653	0.0574 / 0.0586	0.0611 / 0.0602	0.0557 / 0.0569	0.0533 / 0.0570
Stefan-Alternating 150 [dB]	17.16	25.74 / 25.77	24.99 / 25.04	25.70 / 25.67	25.64 / 25.68	25.83 / 25.88	25.88 / 25.86	25.40 / 25.45	25.19 / 25.25
Mode	Fail	D / D	D / D	D / D	D / D	D / D	D / D	D / D	D / D
ME [s]	133	157 / 159	172 / 173	197 / 198	135 / 134	151 / 152	135 / 147	153 / 153	152 / 152
Efficiency index	0.1291	0.1640 / 0.1620	0.1453 / 0.1448	0.1304 / 0.1296	0.1899 / **0.1916**	0.1711 / 0.1703	**0.1917** / 0.1759	0.1660 / 0.1664	0.1657 / 0.1661
Brighter 150 [dB]	18.41	32.99 / 32.40	32.76 / 32.19	33.01 / 32.42	23.80 / 24.63	32.22 / 31.75	32.91 / 32.17	32.05 / 31.54	31.96 / 31.48
Mode	Fail	D / M	D / M	D / M	Fail/Fail	D / M	D / M	D / M	D / M
ME [s]	222	164 / 164	172 / 171	151 / 150	148 / 145	329 / 329	138 / 137	158 / 156	154 / 153
Efficiency index	0.0829	0.2012 / 0.1975	0.1904 / 0.1882	0.2186 / 0.2161	0.1608 / 0.1698	0.0979 / 0.0965	**0.2385** / **0.2349**	0.2029 / 0.2022	0.2076 / 0.2057
SpotLight 100 [dB]	21.81	25.85 / 22.37	26.32 / 22.53	28.10 / 23.21	27.09 / 22.81	28.06 / 23.19	28.13 / 23.22	25.50 / 22.19	27.59 / 23.05
Mode	Fail	B / B	B / B	B / B	B / B	B / B	B / B	B / B	B / B
ME [s]	70	90 / 91	96 / 95	80 / 80	73 / 73	164 / 164	75 / 76	85 / 84	87 / 85
Efficiency index	0.3115	0.2872 / 0.2459	0.2742 / 0.2371	0.3512 / 0.2901	0.3710 / **0.3125**	0.1711 / 0.1414	**0.3750** / 0.3055	0.3000 / 0.2642	0.3172 / 0.2712

remaining rows show the processing time for the motion estimation in seconds and the last row shows the efficiency index which is computed using the following equation:

$$\text{Efficiency index } = \text{PSNR}/T_{\text{ME}} \tag{8}$$

where T_{ME} is the sum of processing times for GME and motion refinement operations.

From the table, it can be seen that the coding quality of the two coding modes are very close to each other for sequences involving no brightness variation or spatially uniform and temporally varying brightness variation (i.e., "Stefan" and "StefanAlternating"). This indicates that both coding modes can handle sequences of these two categories well. However, as nonuniform spatial brightness variation exists ("Brighter" and "SpotLight"), the DCT-based coding mode outperforms the brightness model based coding mode. The difference is particularly significant for "SpotLight" as strong variation of illumination conditions in space exists. In addition to our proposed system, "BlockDivision(Log)," the table also includes an identical coding system "BlockDivision(Ori)" which performs direct division instead of subtraction in log space. From the results, the coding quality of the two systems is very similar to each other while "BlockDivision(Log)" is computationally more efficient as the costly division is replaced. For the DCT-based mode, our proposed "BlockDivision(Log)" coder is the best in terms of the efficiency index for all the sequences involving brightness variations. It outperforms the "Traditional PDS" by 6.32 dB for sequence "SpotLight" as brightness variations exists in space and time domain. After the generation of the sprite image, the video sequence can be encoded into bit-streams for transmission using the MPEG4 VM. Fig. 4 shows the rate distortion diagrams for "SpotLight" using the DCT mode with $N = 10$. From the results, our proposed system significantly outperforms other systems.

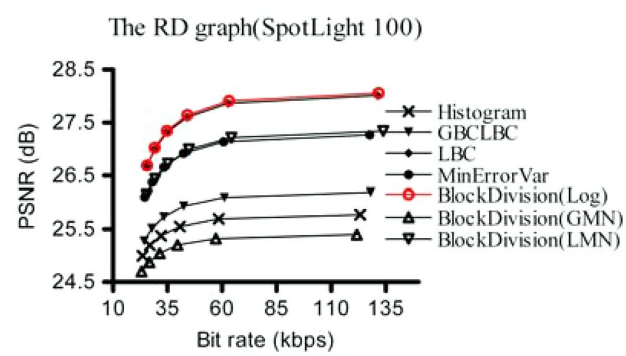

Fig. 4. Rate distortion diagrams for various sequences with different coders using DCT mode and $N = 10$.

V. CONCLUSION

In this brief, we propose a new local motion estimator which can accurately estimate motion activities under varying conditions of strong brightness. Our proposed estimator makes use of a new block division technique which manages practically to get rid of the adverse influence caused by brightness changes between frames. Our proposed estimator is computationally efficient as it avoids the brightness variation estimation and compensation procedure before motion estimation. (This is obviously better than the conventional methods [10], [11] which require the brightness variation estimation and compensation procedure for every candidate macroblock.) The computational complexity can be further reduced by a replacement of the division operations with a subtraction procedure in the log space. Meanwhile, the dimension of the search space is effectively kept to two for the translation model, which is identical to the conventional block-based motion estimators while the computational complexity is also comparable.

To demonstrate the efficiency, we applied our proposed estimator to the static sprite coding application as the coding quality of the system is relatively sensitive to the accuracy of the motion estimator. Our proposed system is also characterized with the capability that the brightness variability of the background scene can be effectively encoded and represented using a single layered sprite image. Experimental results show that our proposed system can efficiently handle video sequences involving different conditions of brightness variations. The performance is particularly outstanding for the sequences involving strong brightness variation in space and time domains in which an improvement of 6.32 dB can be achieved as compared with the conventional static sprite coding system.

REFERENCES

[1] Y. Lu, W. Gao, and F. Wu, "Efficient background video coding with static sprite generation and arbitrary-shape spatial prediction techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 5, pp. 394–405, May 2003.

[2] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 11, pp. 1409–1424, Nov. 2002.

[3] Y. Keller and A. Averbuch, "Fast gradient methods based on global motion estimation for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 4, pp. 300–309, Apr. 2003.

[4] A. Smolic, T. Sikora, and J.-R. Ohm, "Long-term global motion estimation and its application for sprite coding, content description, and segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1227–1242, Dec. 1999.

[5] Y. L. Chan and W. C. Siu, "Edge oriented block motion estimation for video coding," in *Proc. IEE VISP*, Jun. 1997, vol. 144, no. 3.

[6] Y. L. Chan and W. C. Siu, "An efficient search strategy for block motion estimation using image features," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1223–38, Aug. 2001.

[7] S. Negahdaripour, A. Shokrollahi, and M. A. Gennert, "Relaxing the brightness constancy assumption in computing optical flow," *Proc. ICIP*, pp. 806–810, Sep. 1989.

[8] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images," in *Computer Vision Systems*. New York: Academic, 1978.

[9] H. K. Cheung and W. C. Siu, "Robust global motion estimation and novel updating strategy for sprite generation," *IET Image Process.*, vol. 1, no. 1, pp. 13–20, Mar. 2007.

[10] K. Kamikura, H. Watanabe, H. Jozawa, H. Kotera, and S. Ichinose, "Global brightness-variation compensation for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 988–1000, Dec. 1998.

[11] S. H. Kim and R.-H. Park, "Fast local motion-compensation algorithm for video sequences with brightness variations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 4, pp. 289–299, Apr. 2003.