A Video Coding Scheme for Reconstructing an Interest Region with High Quality

Jong Bae Lee and Seong Dae Kim

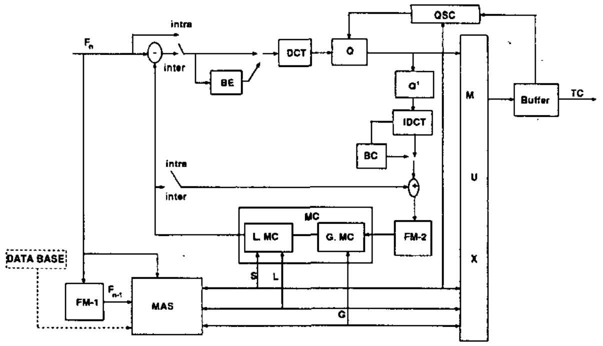
Abstract

In the circumstances we want to deal with, a transmission channel is limited and a global motion can happen by camera movement, and also there exists a region-of-interest(ROI) which is more important than background. So very low bit rate coding algorithm is required and processing of global motion must be considered. Also ROI must be reconstructed with required quality after decoding because of its importance. But the existing methods such as H.261, H.263 can not reconstruct ROIs with high quality because they do not consider the fact that ROIs are more important than background. So a new coding scheme is proposed that describes a method for encoding image sequences distinguishing bits between ROI and background. Experimental results show that the suggested algorithm performs well especially in the circumstances where background changes and the area of ROI is small enough compared with that of background.

I. Introduction

With rapid advances in communications and image processing technology, transmission of digitized video signals becomes a hot issue. To alleviate the channel capacity requirements, it is necessary to employ compression techniques which reduce the data rate without losing too much the subjective quality of images. Here we want to deal with the situation in which a channel is very limited and global motion is likely to happen by camera movement, and also there may exist ROIs which have larger degree-of-importance(DOI) than background. That is, ROI means the degree of a given region indicating 'how important it is' comparing with background(the least important region). So very low bit rate coding algorithm is required and global motion compensation must be done. Also ROI must be regenerated with high quality after decoding because its importance is much larger than that of background.

But the existing methods such as H.261[1] or H.263[2] are not suitable for reconstructing ROI with high quality for following reasons. First, it happens false matching by simple BMA(block matching algorithm) if real motion includes rotation or zooming because velocity vectors are measured by the unit of block(8×8) or macroblock(16×16). Next, it is very difficult to restore ROI because these methods do not consider the fact that ROI is more important than a background. Though the first problems, false matching by BMA scheme, can be solved by global motion compensation[3]-[5], the second problem still remains. In order to solve these problems, we suggest a new method of reconstructing ROI with high quality which segments background and ROI by global/local motion information and codes image with distinguishing bits between ROI and the background. Fig. 1 shows a block diagram of the suggested encoder, which is based on hybrid DPCM/DCT coding with



FM : Frame Memory,
MAS : Motion Analysis and Segmentation
L : Local Motion Parameters,
S : Segmentation Information
G : Global Motion Parameters,
MC : Motion Compensation
FM : Frame Memory,
TC : Transmission Channel

Fig. 1. Block Diagram of Suggested Algorithm.

Manuscript received September 5, 1997; accepted January 13, 1998.

J.B.Lee is with Computer Science Department, Myong Ji Junior College.
S.D.Kim is with the Department of Electrical Engineering, KAIST.

motion estimation and compensation.

This paper suggests two applications. One is that ROI is defined as the region which has a different local motion component(we call this case as UDB from now on). That is, an image is divided into background which has no local motion and ROI which has a local motion. The other is that ROI is predefined and segmentation results can be used to code a given image(we call this case as DB). Although simulation results are suggested for both cases, we concentrate on UDB except buffer control parts because DB is easily applicable by minor modification of UDB.

II. Global Motion Parameter Estimation

BMA scheme estimates x- and y-directional component of motion vector on the assumption that moving object is a rigid body and motion is translational. But motion vectors can not be estimated correctly in the real scenes where moving objects rotate and zooming happens. False matching problems of this scheme are severe in the process of estimating global motion parameters in the circumstances which global motion happens through large area by camera movement. Also if these false motion vectors are used for image coding, image quality is quite degraded after image regeneration and transmission bits are increased by mismatch between current block and motion-compensated previous block. In order to solve these problems, some researchers suggested global motion estimation schemes[3]-[4] based on differential methods. These differential methods use spatio-temporal gradient technique and their performances depend on how accurately the luminance signal can be approximated by the second-order function. In case of large amount of camera motion, these algorithms must iterate many times and often diverge in case of using a few observation points. In order to improve the demerits of these differential schemes, others[5]-[8] suggested an algorithm which can compensate camera zooms and pans effectively by global motion parameters using the local velocity vectors found by the conventional BMAs[9]. Similarly, we suggest a global motion estimation method using four parameters so that we estimate correctly complex motions such as rotation, zooming, and also prevent ROI's or clutters from producing bad effects on these parameters by a refinement process. From [8], we can calculate the estimated value of x- and y-directional velocity components of block b_{ij} in the current image using global motion parameters as listed below.

$$\hat{u}_{ij} = b_1 X_i' + b_2 Y_j' + b_3, \quad \hat{v}_{ij} = -b_2 X_i' + b_1 Y_j' + b_4 \tag{1}$$

where X_i', Y_j' are the x- and y-directional coordinates at a center point of block b_{ij} in current image.

Next local velocity vectors $W_{ij} = (u_{ij}, v_{ij})^T$ are estimated for each block. In order to estimate global motion parameters, let's define l_2 norm of displacement error $E(b_1, b_2, b_3, b_4)$.

$$E(b_1, b_2, b_3, b_4) = \sum_{X_i', Y_j' \in M} \| [(b_1 X_i' + b_2 Y_j' + b_3 - u_{i,j})^2 + (b_2 X_i' - b_1 Y_i' + b_4 - v_{i,j})^2] \|$$

where M is the set of center points for each window.

Finally four global motion parameters are calculated like below by solving above equation.

$$b_1 = \frac{\sum (u_{ij} X_i' - v_{ij} Y_j')}{\sum (X_i'^2 + Y_j'^2)}, \quad b_2 = \frac{\sum (u_{ij} Y_j' + v_{ij} X_i')}{\sum (X_i'^2 + Y_j'^2)}, \quad b_3 = \frac{\sum (u_{ij})}{\sum 1}, \quad b_4 = \frac{\sum (v_{ij})}{\sum 1} \tag{3}$$

where X_i', Y_j' are the x- and y-directional coordinates at a center point of block b_{ij} in current image and u_{ij}, v_{ij} are local velocities in x- and y-directions in block b_{ij} respectively.

Because ROI affects global motion parameters, we suggest a process of refining global motion parameters which iterates two or three times correcting local motion vectors which do not match the previously estimated global motion vectors. If global motion parameters are finally fixed, we can find out the motion compensated previous frame $U_p(x, y)$ using these parameters. That is,

$$U_p(X_i, Y_j) = P(X_i - b_1 X_i - b_2 Y_j - b_3, Y_j - b_2 X_i + b_1 Y_j - b_4) \tag{4}$$

where $P(x, y)$ is previous frame.

III. Region Segmentation

It needs to separate an image into background and ROIs because we should reconsreconstruct them with high quality. In this paper, we suggest an initial background decision which segments background and ROIs roughly using a hierarchical clustering scheme and next, divides the boundaries between background and ROIs more finely using variable blocksize BMA by a quadtree scheme.

In order to determine initial background, we first measure a set of motion vectors for all blocks using full search BMA algorithm between the motion-compensated previous frame $U_p(x, y)$ and current frame $C(x, y)$. Then, we measure the similarity between the velocity vectors of adjacent blocks and define groups. Group is the collection of blocks which have similar velocities. In order to assign the group number to each block, we used the hierarchical clustering algorithm based on the agglomerative clustering method[10]. If cluster groups are determined for all blocks in a given image, initial background can be defined because initial background belongs to the largest cluster group in the image.

In case of undefined regions which are not classified into initial background, further segmentation is done by using the quadtree algorithm[11]. In order to derive quadtree segmentation algorithm, let's first define a rate and distortion because an optimal allocation of rate and distortion is very important for general coding algorithms including ours. From[12], optimal quadtree structure for block X_{n+1} of size $2^{n+1} \times 2^{n+1}$ are found out on the assumption that we know the optimal quadtree structures for four subblocks $X_{n,i}$, $(i=1,\cdots,4)$. Let's put the total rate and distortion of four subblocks as $b^*_{n,i}$ and $d^*_{n,i}$, $(i=1,\cdots,4)$ and also put the rate and distortion given by representing the block with a single leaf at level n+1 as b_{n+1} and d_{n+1}. Four subtrees should be combined into a leaf whenever

$$C_r = \triangle d - \lambda \triangle b \qquad (5)$$

is less than zero where $\triangle d = d_{n+1} - \sum_{i=1}^{4} d^*_{n,i}$, $\triangle b = \sum_{i=1}^{4} b^*_{n,i} - b_{n+1}$ are the distortion increase and rate savings caused by combining four subtrees into a single leaf node.

We suggest the quadtree segmentation algorithm according to the rate and distortion defined above for undefined regions.

<quadtree segmentation algorithm >

Step 1. Let the value of level be zero for the boundaries of an undefined region.

Step 2. Divide these boundary blocks by four subregions and estimate the velocities for each subregion. Increase the level of each divided region.

Step 3. Measure C_r for each boundary block. As mentioned in Eq.(5), $\triangle d$ and $\triangle b$ represent the distortion increase and bit decrease by merging four subregions into one region, and they are measured by Eq.(6) and Eq.(7) respectively.

$$\triangle d = Var(X_{n+1}) - \frac{1}{4}\sum_{i=1}^{4} Var(X_{n,i}) \qquad (6)$$

where $Var(\cdot)$ indicates variance of region $(\cdot)$.

$$\triangle b = \sum_{i=1}^{4} B(Quad(X_{n,i})) - B(Quad(X_{n+1})) \qquad (7)$$

where $B(Quad(\cdot))$ tells total allocation bits for region($\cdot$) when region$(\cdot)$ is represented by the quadtree structure.

If a new group number is assigned to some of subregions, additional representative velocity vectors must be transmitted and it will increase the total allocation bits for four subregions.

Step 4. If C_r is less than zero or the level of a given region reaches maximum value, we finish the quadtree scheme by merging these regions. Otherwise, go to step 2 and iterate again.

IV. Coding of Residual Error Signal and Segmentation Information

In order to represent residual error signals of variable blocks, we use 8 × 8 DCT because suggested method is based on hybrid DPCM/DCT scheme. That is, if a given block is equal to or larger than 8 × 8 block, it is divided into 8 × 8 subblocks and 8 × 8 DCT is done for each divided block. In case of blocks less than 8 × 8 block, it needs to expand these blocks to 8 × 8 block in order to accomplish 8 × 8 DCT. For block expansion, Chang[13] suggested the method of expanding an arbitrarily-shaped region by mirror expansion method and also Cho[14] proposed block expansion by 1-D interpolation in frequency domain and time domain respectively. Our block expansion scheme[6] basically follows Cho's method. But in case of 4 × 4 blocks, more efficiently, 2-D interpolation is possible. Table 1 shows the average bits and PSNR of 2-D interpolation, zero-stuffing and mirror-expansion method from first frame to fifth frame of table tennis sequence, which shows 2-D interpolation has the best performance, mirror expansion has the second and zero-stuffing has the worst performance in considering bits and PSNR simultaneously.

Table 1. Simulation results for block expansion (PSNR(dB)/used bit) Table tennis sequence

TYPE	Zero stuffing	Mirror	Interpolation
Fig. 2(a)	22.33/125054	29.10/46149	32.68/44014
Fig. 2(b)	26.05/118501	29.29/68931	31.17/66946
Fig. 2(c)	24.03/128567	27.11/66047	27.37/64598

In order to represent segmentation information between background and ROIs, we can use several coding algorithms such as chain coding, run length coding or quadtree scheme. In this paper, quadtree scheme by depth first search method is used among them because of following reasons. First, quadtree scheme is more easily realizable because quadtree scheme is previously used for segmentation. Next, the number of total bits used to represent segmentation information in quadtree scheme are smaller than that in chain coding or in run length coding. The main reason is that there exist many separate regions in our algorithm which results in additional bit increase in order to represent initial position of each region in chain coding or do several labels and runs in run-length coding. Fig. 2 represents the generated bits by three methods for table tennis sequence.

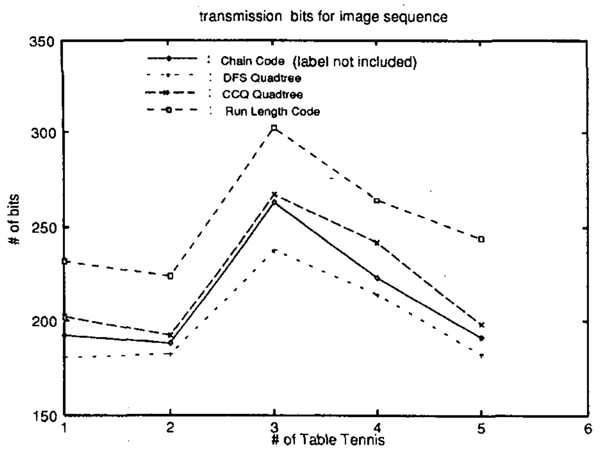


Fig. 2. Total bits for segmentation information.

VI. Buffer Control

In the existing schemes such as MPEG, H.261, quantization step size is regulated in the unit of macroblock, GOB(group of block), or GOP(group of picture), without considering the importance between ROIs and other regions. But it sometimes needs to use DOI in order to regulate quantization step size and buffer state. For example, chairman or speakers who moderate a conference are more important than the background in a video conferencing system, and degradation of these parts has worse effect on users than that of other parts.

Another example is the application of tactical scene : airplane or fighter has more useful information than the clutters such as cloud, sun or birds in an anti-aircraft detector, and the movement of tank, armored motorcar of soldiers is more important than that of trees or reeds.

1. *Quantization parameter decision*

Quantization parameter must be precalculated in order to quantize DCT coefficients for error signals for several regions whose DOIs are different. Let's define that $E_{\max}$ is the maximum admissible error for ROIs and $\Theta_T(b_i)$ the average quantization error of the i-th block which belongs to ROI. We must assign bits b_i according to Eq.(8) in order to reconstruct ROI with required quality.

$$\frac{1}{B_T}\sum_i \Theta(b_i) \le E_{\max} \qquad (8)$$

where B_T is the number of blocks which belong to ROI.

Also the number of quantization bits for background error signals are determined by Eq.(9) if R indicates the overall system rate.

$$b_s \le \frac{R}{frame\ \ rate} - \sum_{b_i \in R_i} b_i - P_a \qquad (9)$$

where b_s means the bit rate that can be used to transmit the background signal and P_a several parameter information which includes global motion parameters, velocities of groups and position information, etc.

2. *Buffer control*

In this subsection, we will treat only the buffer control of DB because UDB is considered as a special case of DB which consists of two regions; one is background and the other is ROI. Fig. 3 briefly shows the process of buffer control in case that there exist three regions having different DOIs such that r_a means the most important region, r_b the second, and r_c the third. X-axis indicates the size of the region which has the largest DOI(r_a) and Y-axis indicates PSNR of each region according to the change of x value. In fig. 3, r_a and r_b keep optimal quality within the range of $t_1 \ge 0$. That is, these two regions can be controlled optimally within the buffer capacity. In case of background, quality is continuously degraded. As the value increases above t_1, r_a still keeps constant quality, but r_b starts to degrade until t_2 level. After t_2 level, the quality of r_a does not maintain the optimal quality any more and it also deteriorates continuously until x values reach t_3 level. That is, r_c reaches the minimum state at t_1 level, and region r_b at t_2 level , and r_c at t_3 level, respectively. Therefore this scheme first keeps the quality of most important region constant and optimal, and next the other regions according to DOI and remainder bits.

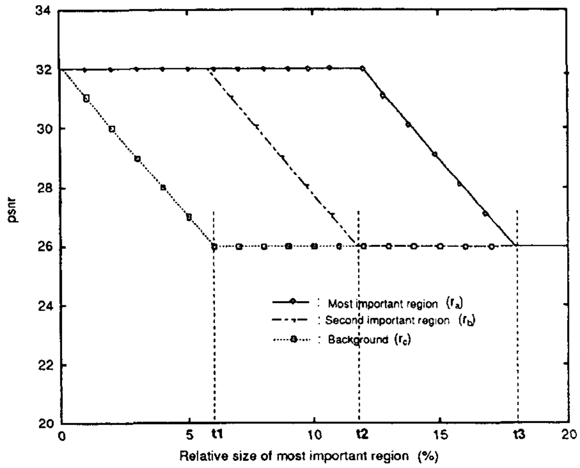


Fig. 3. Performance according to the size of most important region.

Let's define some notations before proceeding.

$bpp_{\min}(r_i)$: minimum bits/pixel for region r_i
$bpp_{opt}(r_i)$: desired bits/pixel for region r_i
R_S : side data such as motion and segmentation information
a : inter-block, b : intra-block
$\min_{r_i}(=\sum_{k=i}^{k=n} bpp_{\min}(r_k^a + r_k^b))$: overall bit in order to satisfy minimum quality of region $\{ r_k,\ k=i,\cdots,n \}$
$\max_{r_i}(=\sum_{k=i}^{k=n} bpp_{opt}(r_k^a + r_k^b))$: overall bit in order to satisfy desired quality of region $\{ r_k,\ k=i,\cdots,n \}$
R_{r_i} : Bits used to code region r_i, $R_t = R/(frame\ rate)$: target bits
R : bit rate per second, BF_{init} : initial buffer state
$BF_{pre}(=BF_{init} + \sum_{k=1}^{k=i-1} R_{r_k} - (i-1)R^t)$: buffer state after the completion of until previous frame
$VAR(r_i)$: variance of region r_i

Frame rate for current frame R_{cur} is defined as Eq.(10). That is, the bits can be used to code current frame is sum of R_t and BF_{init} subtracting from BF_{pre}.

$$R_{cur} = R_t + (BF_{init} - BF_{pre}) \tag{10}$$

Let's investigate buffer control process step by step. We first assume that DOI of each region is like this.

$$M_{r_1} > M_{r_2} > \cdots > M_{r_i} > \cdots > M_{r_n} \tag{11}$$

where $M(\cdot)$: DOI of region $(\cdot)$

Coding process for region r_i is divided into three cases according to the remainder bits which can be used.

Case 1 : $\min_{r_i} < R_{i_{remain}} < \max_{r_i}$

Most states correspond to this case and the quality of region r_i can be kept at least minimum level, and it is also divided into three types.

a) $bpp_{opt}(r_i^a) + bpp_{opt}(r_i^b) > R_{i_{remain}} - \min_{r_{i+1}}$ and
$bpp_{opt}(r_i^a) + bpp_{opt}(r_i^b) < R_{i_{remain}} - \max_{r_{i+1}}$

Here, the estimation of bits for region r_i in current frame, $\widehat{bpp}_{opt}(r_i^a)_{cur}$ is calculated by Eq.(12).

$$\widehat{bpp}_{opt}(r_i^a)_{cur} = bpp_{opt}(r_i^a)_{pre} \times \frac{Sizeof(r_i^a)_{cur}}{Sizeof(r_i^a)_{pre}} \times \frac{VAR(r_i^a)_{cur}}{VAR(r_i^a)_{pre}} \tag{12}$$

where $(\cdot)_{cur}$ means region $(\cdot)$ of current frame and $(\cdot)_{pre}$ means region $(\cdot)$ of previous frame.

This type is the case that region r_i can be reconstructed with optimal quality. Optimal QP value is used in order to satisfy desired quality of region r_i.

b) $bpp_{opt}(r_i^a) + bpp_{opt}(r_i^b) < R_{i_{remain}} - \min_{r_{i+1}}$

This type is the case that remainder bit is not sufficient to keep optimal quality for region r_i. So QP value is adaptively increased in order not to exceed overall bit allocated in region r_i.

c) $bpp_{opt}(r_i^a) + bpp_{opt}(r_i^b) > R_{i_{remain}} - \max_{r_{i+1}}$

This type is the case that more bits are left after coding region r_i with optimal quality. So QP value is adaptively decreased in order to remove remainder bits as Eq.(13).

$$\triangle QP \simeq k \ \triangle x \tag{13}$$

where k is a constant, $\triangle x$ indicates remainder bits.

Such a coding, generally, does not make buffer overflow. That is,

$$BF_{ov} < BF_{prev} - BF_{init} + R_S + \sum_{k=1}^{i} R_{r_k} \tag{14}$$

But especially if present buffer state reaches BF_{ov} after coding region r_i, we should prevent buffer overflow by increasing quantization level adaptively and coding region r_i again with this quantization parameter.

Case 2. $R_{i_{remain}} > \max_{r_i}$

This type is the case that more bits are left after coding remainder regions $r_k, (k=i,\cdots,n)$ with optimal quality. So QP value is adaptively decreased in order to remove remainder bits like Eq.(13).

Case 3. $R_{i_{remain}} < \min_{r_i}$

This type is the case that all remainder regions $r_k, (k=i,\cdots,n)$ can be reconstructed even with minimum quality. If $R_{i_{\min}} > \min_{r_i}$, $R_{i_{\min}}$ is coded in selecting QP value which can satisfy minimum quality of region r_i. Otherwise, we do not code $\sum_{k=1}^{k=i} R_{r_k}$ any more and finish coding process of this frame.

VI. Experimental Results and Concluding Remarks

Conditions of simulation are defined as follows. In case of Table tennis sequence, we set picture size as 352 × 288, bit rate as 32 kbits/s and frame rate as 30 frames/s for monochrome image. Also in case of Trevor sequence, we set picture size as 352 × 288, bit rate as 48 kbits/s and frame rate as 10 frames/s for monochrome image. Fig. 4 (a) and (b) show motion vectors of suggested algorithm before and after global motion compensation for Table tennis sequence, respectively. The results show that global motion compensation must be done in order to represent camera movement well and reduce the bits to transmit motion vectors. Fig. 5 - Fig. 8 represent the result of UDB. Fig. 5 describes a reconstructed image which represents the original image well, especially a pingpong ball, which is set as ROI. Fig. 6 represents the ROI reconstructed by suggested method, and Fig. 7 (a), (b) and (c) show PSNR of image sequence, PSNR of ROI, and transmission bits respectively for Table tennis sequence. The PSNR of full search BMA scheme (FBMA) used in H.261[1] or H.263[2] is worse than the other methods because zoom component is not corrected well. BMA scheme after global motion compensation(GFBMA) has higher PSNR value than FBMA for both background and ROI and its PSNR is similar to that of suggested method (SGFBMA) for background because zoom is globally compensated. But PSNR of GFBMA for the ROI is still lower than that of SGFBMA and is not be able to reconstruct the ROI with required quality. But suggested scheme needs more computational efforts than the other schemes because SGFBMA has more additional functions than FBMA or GFBMA. Fig. 8 shows the simulational result of trevor sequence, which also shows that our selective scheme represents ROIs with better quality than non-selective scheme.

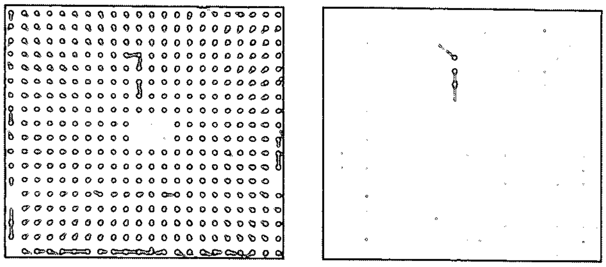

(a) (b)

Fig. 4. Estimated motion vector (a) Before global motion compensation (b) After global motion compensation.

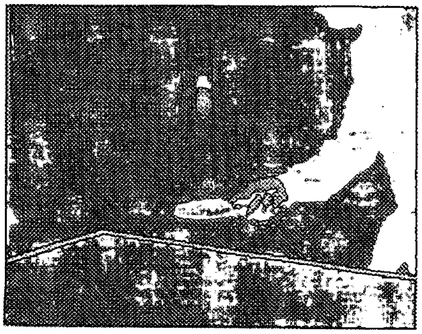


Fig. 5. Reconstructed Image.

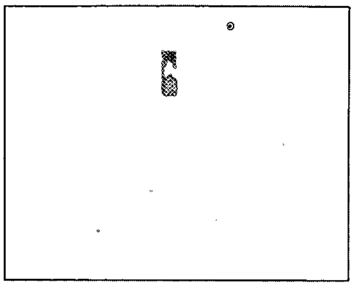
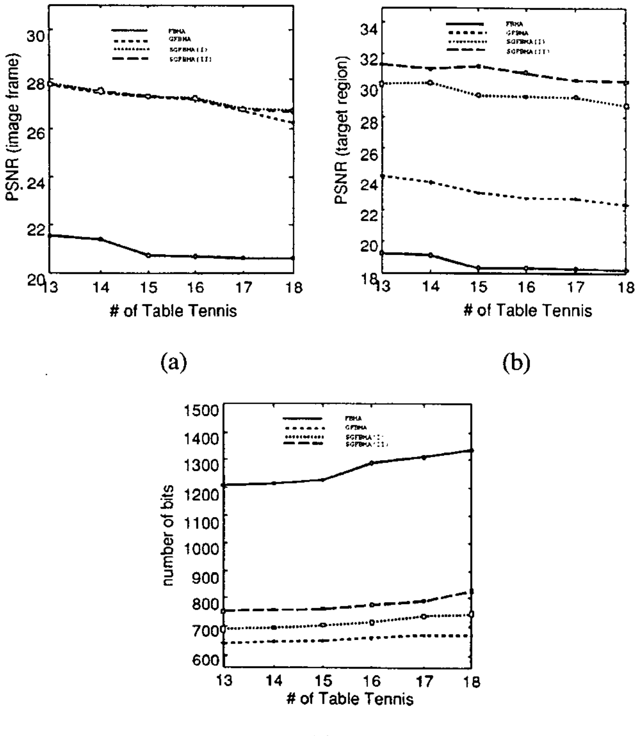


Fig. 6. Reconstructed ROI.



(a) (b)

(c)

Fig. 7. PSNR and transmission bits for image sequence (a) Performance of image seq. (b) Performance of interest region. (c) transmission bits for image seq.

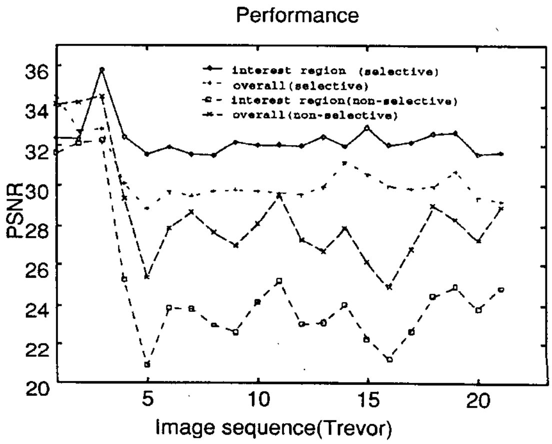


Fig. 8. Performance of Trevor sequence(48 kbps 10 frame/s)

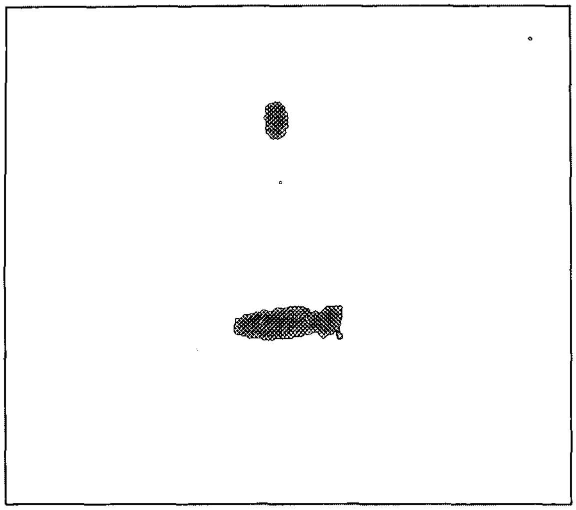


Fig. 9. Example of ROIs for DB.

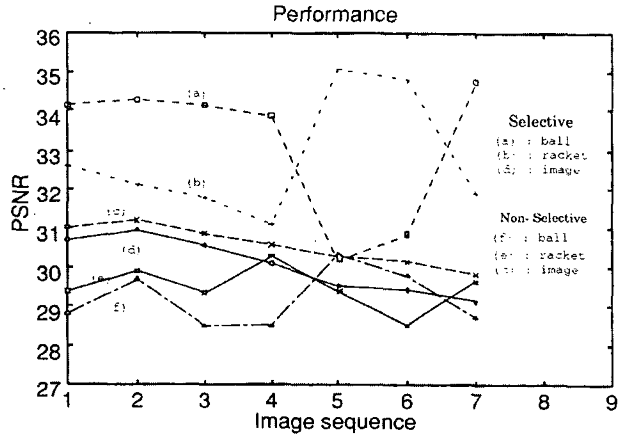


Fig. 10. Performance of DB simulation for selective and non-selective coding.

We also have simulated in case of DB that ROIs are predefined like Fig. 9. Fig. 10 represents the result, which also shows that selective coding has better performance.

In conclusion, our algorithm has higher PSNR gain and smaller transmission bits compared with other two algorithms because local motion vector information is very little by grouping similar motion vectors through background/ROI clustering and quad-tree segmentation. Also in our algorithm ROI has much higher PSNR values than background because sufficient bits are allocated to ROI in order to reconstruct the ROI with required quality.

References

[1] "Draft revision H.261," *CCITT SGXV*, March 1990.

[2] "Draft Recommendation H.263", ITU-TS SG15 Working Party 15/1, March 1995.

[3] M. H ötter, "Differential estimation of the global motion parameters zoom and pan," *Signal Processing,* Vol. 16, 1989, pp. 249-265.

[4] S.F.Wu and J.Kittler (1989), "A differential method for simultaneous estimation of rotation, change of scale and translation", *Signal Process.* , pp. 69-80.

[5] Y. Tong Tse and R.L.Baker (1991), "Global Zoom / Pan Estimation and Compensation for video Compression", *Proc. ICASSP - 91*, pp. 2725-2728.

[6] J. B. Lee, "Selective Coding Algorithm based on motion vector information", Ph. D. thesis, Feb. 1996.

[7] J. B. Lee and S. D. Kim, "Moving target extraction and image coding Based on motion information", *IEICE Trans. Fundamentals*, Vol. E78-A, Jan. 1995, pp. 127-130

[8] J. B. Lee and S. D. Kim, "Target Region Extraction and Image Coding Based on Motion Information", *VCIP-95*, May 1995, pp. 705-716.

[9] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation", *IEEE Trans. Commun.*, Vol. 33, 1985, pp. 888-896.

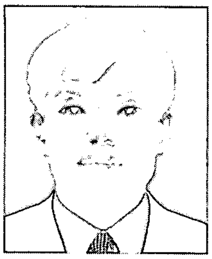
[10] R. O. Duda and P. E. Hart (1973), Pattern Classification and Scene Analysis, *Wiley-Interscience.*

[11] D. H. Ballard and C. M. Brown (1982), Computer Vision, *Prentice Hall.*

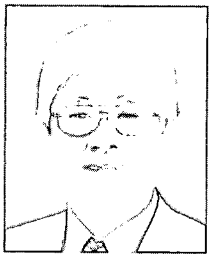
[12] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of image and video", *IEEE Int. Conf. on ASSP*, May 1991, pp. 2661 - 2664.

[13] S. F. Chang and D. G. Messerschmitt, "Transform Coding of Arbitrary-Shaped Image Segments", *ACM Multimedia,* Vol.1, 1993, pp.83-90.

[14] S. J. Cho and S. D. Kim, "Texture coding using 2-D DCT based on extension interpolation", *IEICE Trans. Fundamentals*, Vol. E80-A, pp.789-793, 1997.



Jong-Bae Lee received the B.S. degree in Control and Instrumentation engineering in 1985 from Seoul National University, and the M.S. and Ph.D. degree from KAIST(Korean Advanced Institute of Science and Technology), in 1987 and 1996, respectively. He is a professor in Myong Ji Junior College. His current research interests include image processing. pattern recognition, and image coding.



Seong-Dae Kim received the B.S. degree in electronic engineering in 1977 from Seoul National University. and the M.S. degree from KAIS(Korea Advanced Institute of Science), Seoul, in 1979. He received the Diplome de Doeteur-Ingenieur from Ecole Nationale Superieure d' Electrotechnique'd Electronique d'In formatique et d' Hydraulique (ENSEEIHT) of the Institute National Polytedmique de Toulouse(INPT), Toulouse, France, in 1983. He joined the department of Electrical Engineering at the KAISt in September 1984, where he is currently a professor. His current research interests include innage processing, image coding, HDTV, hardware realizations of image processing systems. He is a member of the Korean Institute of Telematics and Electronics(KITS), the Korean Institute of Communication Sciences (KICS), and the IEEE.