This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Hybrid Object-Based Video Compression Scheme Using a Novel Content-Based Automatic Segmentation Algorithm

N. A. Tsoligkas, D. Xu and I. French

School of Science and Technology, University of Teesside, Middlesbrough, TS1 3BA, UK
E-mails: tsoligas@teihal.gr, d.xu@tees.ac.uk, I.french@tees.ac.uk

Abstract - **This paper describes a hybrid object-based video coding scheme that achieves efficient compression by separating moving objects from stationary background and transmitting the shape, motion and residuals for each segmented object. In this scheme, a new content-based object segmentation algorithm is proposed, which does not assume any prior modeling of the objects being segmented. The binarization process, which finds large object regions, is based on a threshold function that calculates block histograms and takes image noise into account. The resultant binary mask is further processed using morphological operations. The motion vectors are estimated inside the change detection mask using block-matching method between two successive frames, and then the dense motion field is estimated using the motion vectors and the Horn-Schunck algorithm.**

Keywords - **Content-based video coding, object segmentation, motion estimation, motion compensation, motion failure area, variable length coding.**

I. INTRODUCTION

The video coding used in modern multimedia communications employs the techniques that use intra-frame coding and inter-frame coding to achieve high compression ratio. The first technique operates on each frame of a video sequence while the second exploits temporal correlation between frames, thereby achieving much higher compression efficiency. With emerging wireless and mobile applications, there has been considerable research in the area of the object-based coding for very low bit rate compression. The compression is achieved by separating coherent moving objects from stationary background and compactly representing their shapes, motions and contents [1]-[7]. Also is known that the object-based coding techniques can support content-based functionalities such as to code and decode and in general manipulate specific objects in a video stream. However, most of the object-based coding techniques have two major drawbacks. Firstly, object segmentation and motion estimation methods are computationally very intensive. Secondly, accurately representing the shapes of moving objects would result in insufficient number of bits for coding the content of a low bit rate encoder. Although the shape coding can be avoided by using a fixed block-based partitioning technique [8], such as the overlapped block motion compensation, which also help reduce prediction error significantly. But in general the block-based coding suffers from the use of a fixed block partitioning on a fixed grid, which results in blocking artifacts.

In this paper, a hybrid object-based video coding scheme is presented, incorporating a new object segmentation tool that retains the characteristics of both object-based and block-based coding. The object segmentation technique is based on the generation of a binary mask of the objects being segmented. The position of an object is unknown and has to be determined based on its motion. Two successive frames are subtracted to generate a change detection mask [9]-[10], and then the shape, motion and residuals of each object are coded accordingly. Content-based functionalities, e.g. scalability, can be supported by selectively encoding and decoding objects in a video stream.

The rest of this paper is organised as follows. The proposed hybrid object-based coding method is explained in detail in section II, including object segmentation, motion estimation, motion failure area, and residual coding, etc. In section III, the experimental results and the performance comparison between conventional and the proposed methods are presented. Finally, conclusions are drawn in section IV.

II. STRUCTURE OF HYBRID OBJECT-BASED CODING

A. Overview

The block diagram of the proposed hybrid object-based video coding scheme is shown in Fig.1 [11]-[13]. It operates, basically, by estimating moving objects and coding compact representations of the objects shape, motion and residual signal. The first frame of a video sequence is intra-frame coded using DCT transform, quantization, run-length coding and Huffman coding. The following frames are then coded with the processes below.

- Objects motion detection;
- Motion estimation and compensation;
- Shape representation and coding;
- Motion failure region detection;
- Frame reconstruction;
- Residual encoding.

1-4244-0353-7/07/$25.00 ©2007 IEEE

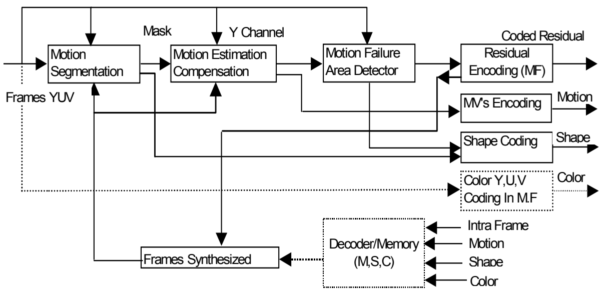

Figure 1. Hybrid object-based coding

The modules in the hybrid object-based coding scheme can be easily changed as new techniques/algorithms are developed.

B. Object Motion Detection - Segmentation

The purpose of the change detection unit is to distinguish between two successive frames the temporally changed/unchanged regions. Here the selection of an appropriate threshold is key because a low threshold may cause over-segmentation, while a high threshold could result in under-segmentation and incomplete objects. Obviously, a large moving region requires a large number of bits for its representation. There have been many techniques developed to determine a threshold for binarization of an intensity image [14]-[15]. Most of them are based on the histogram of the intensity signal. However, a difference image differs from intensity images, and the threshold methods for the intensity images may not be appropriate to be used for difference image. While block-based coding techniques divide an image into a set of square regions, object-based schemes divide an image along objects boundaries. Since it is not necessary to produce a precise segmentation of the moving objects in terms of gaining high coding efficiency, the motion detection and segmentation method in Fig.1 uses the statistical properties between two successive frames, which include three major steps as follows.

B1. Image differencing

The objective of differencing image is to identify the portions/areas of image, which have changed from one frame to the next. This is accomplished by computing the absolute value difference of two images pixel by pixel.

B2. Image filtering

The absolute difference image normally includes artifacts due to illumination and noise changes. By averaging the difference, using a 5x5 kernel average filter, the image becomes smooth while uncorrelated noises are reduced. Stability around the object boundaries and limitation of the motion detection to the neighborhood of the current pixel are improved by the use of a maximum filter [16]-[17].

B3. Image threshold

Threshold computation and image binarization are designed to be tolerant of large variations in image intensity and contrast. Knowing that the thresholding methods can be divided into two categories, global and local, we combine the global method (block) with the local method (block-histogram-based) to calculate threshold. Assume that an image is divided into B equal blocks with the size of each block being $R{\times}C$, where R and C are the numbers of rows and columns, the threshold Th is calculated as follows.

$$T_h = \frac{1}{((2 \mathrm{x} B)+B)} \mathrm{x} \sum_n^B (\sum_i^2 (m_i)+\mu_\eta) \tag{1}$$

$$\mu_b = \frac{1}{R \times C} \mathrm{x} \sum_i^R \{\sum_j^C (FD(i,j))\} \tag{2}$$

where m_i is the clustering mean and μ_b is the average gray level of the block. The intensity histogram of each block is divided into two or N clusters. Using the N values of the clustering mean intensity m_i of each block, the threshold T_h is calculated by averaging all the means m_i, and all the μ_b for all the B blocks, (or we can use an equidistant-bin histogram of each block, and for each bin the most frequent level m_i is fixed). In order to adapt to image noise, the threshold is further expressed by

$$T_t = T_h + k \times \sigma^2 \tag{3}$$

where σ^2 is image noise variance [18] and k is a constant less than 1. In order to determine the final threshold, T_t is compared with the previous calculated threshold T_{t-1}. If the current threshold T_t is greater than the previous threshold T_{t-1}, T_t is selected; otherwise, $Tt\text{-}1$ is selected. When a small motion or no motion is detected (threshold is below a certain value) T_t is calculated by taking the average value of all the previous threshold values. Thus, the detection of the binary object mask is stabilized. Each pixel of the frame difference $FD(i,j,t)$ is classified as either belonging to an object and labelled white in a binary image mask, $M(i,j,t)$ or belonging to the background and labelled black. The threshold computation and binarization process is illustrated in Fig.2.

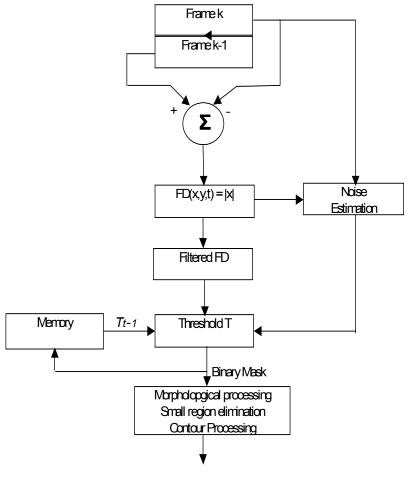

Figure 2. Flowchart of threshold computation and binarisation

The binary mask resulted from the binarization process may contain artifacts, especially around objects boundaries, so a post-processing step is applied to the binary image. First a median with 3x3 kernel filter is used, and then three successive morphological closing operations with 3×3 kernel, followed by three morphological opening operations with the same kernel are applied to clean up raw intensity differences and then to cluster moving objects regions. Small regions (smaller than 100 pixels) are eliminated. Fig. 3 shows the 16th frame of the 'Akiyo' video sequence, and Fig. 4 shows the 15th synthesized frame. Examples of the detected moving regions and the motion failure regions after motion compensation are shown in Fig. 5 and Fig. 6, respectively.

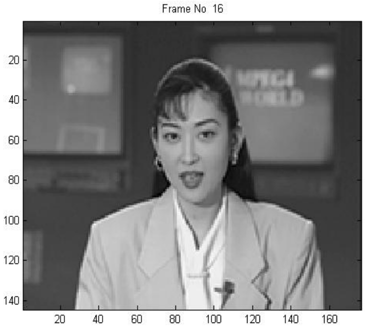

Figure 3. Original current frame

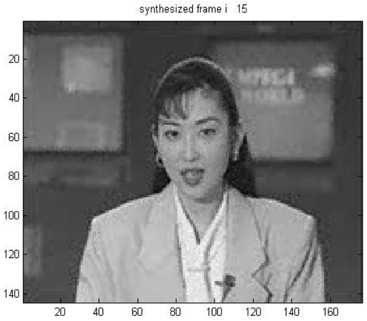

Figure 4. Previously reconstructed frame

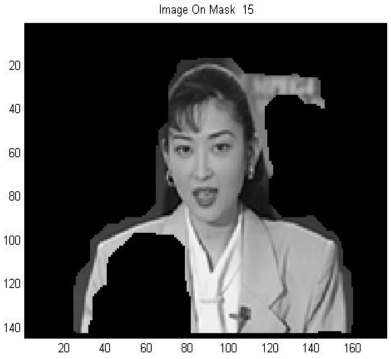

Figure 5. Detected moving areas

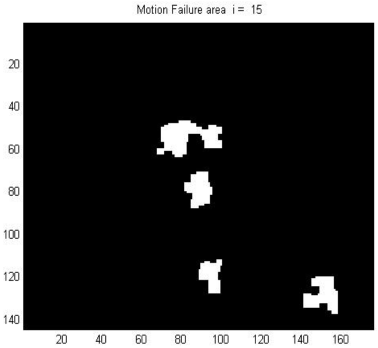

Figure 6. Motion failure areas

It is clear that the segmentation does not exactly follow the boundary between the foreground and the background. Better performance in terms of boundary refinement can be achieved if other moving object detection techniques are used [19]-[22]. Here, the produced binary mask is superimposed on the current frame to extract the object in motion and reject the rest of the image. Advantage of this method is that if camera is panning, this motion can also be detected and compression will therefore be achieved as a result of the subsequent steps.

C. Motion estimation and compensation

To remove temporal redundancies of a video, motion estimation needs to be performed to predict the contents of the segmented regions based on previously synthesized frame. Each macroblock in current frame is compared with the macroblocks within a search range in the reference (previous) frame by measuring the error through computing the sum of absolute differences (SAD). The best matching macroblock is selected. For the current frame, this macroblock can be encoded as a motion vector, which denotes only the translational displacement of the macroblock in the reference frame in new position. In order to improve estimation accuracy, two approaches have been considered here. The first is to widen the search range and to use smaller macroblock size. This method obviously increases computation significantly. The second technique is so-called fractional-pixel motion estimation (quarter-pixel or half-pixel) [23], which is adopted in the scheme proposed. After motion estimation is carried out, motion vectors are predictively coded using fixed predictor coefficients, and the prediction errors for the displacement vectors undergo Huffman coding. Now the block correlation is performed inside the change detection mask (CDM) and the resulting coarsely sampled motion vector field is interpolated to a dense field and passed through the iterations of the Horn-Schunck algorithm [24]. This process smoothes the vector field used to predict the frame.

D. Motion failure

Motion failure region detection refers to clusters of pixels in an area where motion compensation alone was inadequate. The boundaries of these regions can be estimated by thresholding the displaced frames difference, computed from the forward dense motion field estimation, i.e., from the frame k to k+1. Clearly, the accuracy of these boundaries depends on the accuracy of dense motion estimation. Here, optic-flow equation based method, such as Horn-Schunck method [24], is used. A further motion segmentation is performed between the current frame and the motion compensated frame. The threshold used is defined by:

$$\begin{aligned} &|f_{k+1}(x,y) - f'_{k+1}(x,y)| < T_{MF} \ \ and \\ &T_{MF} = c \times \frac{1}{N}\sum_{n}^{N} |f_{k+1}(x_n, y_n) - f'_{k+1}(x_n, y_n)| \end{aligned} \tag{4}$$

where the subscript *n* denotes the index of pixels forming the moving area, *N* is number of the pixels included in the moving area, $f'_{k+1}(x,y)$ denotes the intensity value of the *(k+1)*th reconstructed frame, and *c* is a constant.

The number of bits required for the motion failure (MF) region coding is usually large, so the extracted area should be relatively small. The minimum size of the motion failure region is normally set to 0.2% of the total number of pixels in an image. For the QCIF 'Akiyo' sequence with frame size 144×176, the minimum size is about 50 pixels. The regions with a size smaller than 50 pixels are eliminated for coding efficiency. The average size of the motion failure area over an entire frame is normally set to 5% of the total number of pixels in an image, e.g., 1300 pixels in the above example. Therefore, the amount of bits required for coding is actually controlled. Examples of motion failure regions detected using this technique are shown in Fig. 6. The shapes of these motion failure areas are coded by polygon or cubic smooth spline approximation.

E. Residual encoding

The residual information inside the motion failure regions is used to correct the errors produced during the motion estimation. The coding techniques applied include the discrete cosine transform (DCT), the wavelets transform, the shape adaptive DCT (SADCT), and vector quantization (VQ) [7], etc. Typically, the pixels in the motion failure area are coded by the DCT transform, which is performed on 8×8 pixel blocks. The quantized DCT coefficients are then run–length encoded in a zigzag fashion and finally Huffman coding is applied [19].

The wavelets coding approach consists of a wavelet decomposition of the entire error image (five level decomposition for the luminance pixels), followed by quantization and Huffman coding. In our implementation, a bi-orthogonal filter has been used. The synthesized image is passed through a spatial filter before its subtraction from the incoming video sequence. Compared to DCT technique, the quality of the synthesized image is improved.

F. Shape analysis

The boundaries of the detected moving regions need to be approximated by a shape model that can be represented with a few parameters. Most commonly employed shape models are:

- A bounding rectangle for each of the moving regions, which is tiled with 16x16 pixel macroblocks [25]. For transmission purpose, only the parameters like row-column position vector, the height and width of the shape in macroblocks, and run length encoded shape mask are transmitted.
- A polygon that is used in our case. The polygon approximation algorithm [3]-[7] and cubic smoothing spline data interpolation are adopted to approximate shapes.

The traced contours are differentially coded and then run-length coded. In both encoding and decoding, the contours are approximated by fitting splines to the corner points. While applying this algorithmic procedure to motion failure object shapes, the number of vertices may be reduced (sampled), and therefore the simplification of shape can be introduced. If the objects shapes are interpreted as a set of blocks, the analysis-synthesis codec technique is the same as the hybrid codec technique.

III. EXPERIMENTAL RESULTS AND ANALYSIS

To test the scheme/algorithms proposed, the simulation and experiments are carried out using the test sequence Akiyo, which is a Class-A sequence. This test sequence is characterized as having slow motions and simple spatial details, typical for videoconferencing and videophone applications. Fig. 3 and Fig. 4 show an 'original' video frame of this video sequence and a reconstructed frame using the codec shown in Fig. 1. Fig. 5 shows the change regions resulted from applying the moving object detection techniques described in the section B3 and from the image thresholding equations (1), (2) and (3). The motion failure objects shown in Fig. 6 are detected by applying the threshold values of (4).

The criterion used to measure reconstructed image's quality is the PSNR (peak signal to noise ratio) that is defined by:

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{(1/(W \times H)) \sum \sum [f(x,y) - f'(x,y)]^2}$$

where *W×H* is image size, and *f(x,y)* and *f'(x,y)* are the pixel intensities of the original and reconstructed images at the coordinate point *(x,y)*, respectively. The luminance and chrominance components are predicted using the same set of motion vectors. So no additional overhead is required to predict the chrominance. The motion vectors are estimated using the luminance information only. However, the chromo mismatch correction for the chromo blocks has not been applied here. The first frame of the sequence is intra-frame coded and decoded and used as the first synthesized frame. In order to demonstrate the performance of the proposed coding scheme, Fig. 7 shows PSNR of the three different coding methods (low-bit coding, DCT and wavelets), all of which include moving object detection, motion estimation and motion failure region detection techniques.

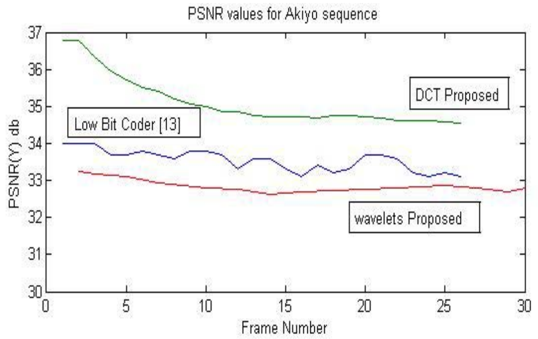

Figure 7. PSNR of three coding methods

It is worth mentioning that PSNR of the proposed DCT method is 2.8dB higher on average than the conventional one.

Table 1 gives the performance comparisons between the proposed scheme and the Schiller [26], Wollborn [27] and Mukherjee [28] methods in terms of the average number of bits required for coding one frame and PSNR.

TABLE 1
PERFORMANCE COMPARISONS BETWEEN CODING METHODS

Scheme	Number of bits	PSNR (dB)
Schiller [26]	11180	34.06
Wollborn [27]	10290	34.77
Mukherjee [28]	13400	35.20
The proposed	**12644**	**35.10**

In the proposed codec, the number of average bits required for coding one frame is similar to or slightly higher than that of other schemes. This is because in the shape analysis, encoder detects the corner points along contours and sends these to the decoder that uses a differential coding technique, and also the maximum number of corner points, rather than the reduced one, has been used for decoding in our experiments. The decoder approximates the contours by fitting splines to the corner points so that both encoder and decoder agree precisely. The experiments relating to picture quality and sub-sampling corner points will be undertake in the next stage of the work. The proposed scheme yields a marginally higher PSNR than other methods with the similar number of bits. However, the algorithm to find the Minimum Perimeter Polygon (MPP) [7] of a region has not been applied yet. With use of the MPP, it is expected that PSNR will improve significantly.

As discussed in the section II, the motion failure (MF) objects are extracted through the threshold of (4). In our experiments, the average size of an MF object is set to about 1300 pixels for a QCIF format (144×176) image. The resultant percentage of the motion failure area to the total image area is shown in Fig.8, with a mean value of 2.4%. This is in correspondence with that when PSNR is controlled to be stable around 40dB, the MF area should be kept below 3% of the total image area [29].

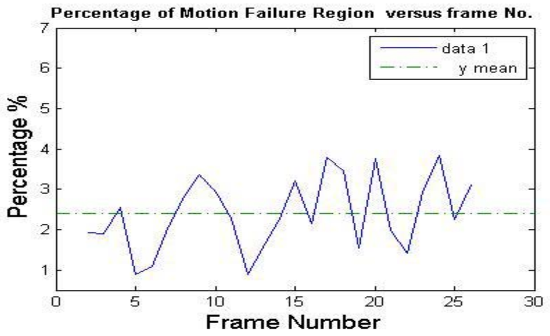

Figure 8. Percentage of MF area to total image area

The proposed hybrid object-based coding scheme, needs 7.2 seconds on an Intel P4 2.8 GHz processor with 2 GB ram PC to read 100 frames. 3.9 seconds to code and decode the first frame and 23.4 seconds per frame for motion detection, thresholding, noise estimation, morphological operations, shape analysis, filtering, coding and reconstructed the frame.

IV. CONCLUSIONS

A hybrid video coding scheme that combines traditional block-based method with object-based coding has been presented. It aims primarily at low bit rate transmission applications (typically below 16 kbits/s). In the segmentation algorithm that is developed to be robust to additive noise, both illumination changes and noise are taken into account by way of dividing image into blocks and clustering the histogram of each block into two clusters. By using a median filter in the segmentation, smooth and stable object boundaries are produced.

The block correlation performed inside the CDM generates a coarse sampled motion field, which is then interpolated to a dense field by passing through 15 iterations of the Horn-Schunck algorithm. The resulted smooth dense motion estimation is therefore used for motion compensation to provide high prediction accuracy. Using spline-based shape representation (or bounding rectangles), the scheme also achieves improved coding efficiency compared with the classical object-based coding.

The modular nature of this coding scheme enables the development and use of other coding techniques, e.g., wavelet coding techniques, shape adaptive-DCT and vector quantization in the residual encoding module, thus providing flexibility and the room for further improvement.

REFERENCES

[1] H. G. Musman, P. Pirsch, and H.J. Grallent, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523-548, April 1985.

[2] R. Schafer and T. Sikora, " Digital video coding standards and their role in video communications," *Proc. IEEE*, vol. 83, No. 6, pp. 907-923, June 1995.

[3] M. Hotter and R. Thoma, " Image segmentation based on object oriented mapping parameter estimation", *Signal Processing*, vol. 15, pp. 315-334, 1998.

[4] M. Hotter, " Optimization and efficiency of an object-oriented analysis-synthesis coder," *IEEE Transactions on Circuits and Systems for video technology*, vol. 4, pp. 181-194, April 1994.

[5] H. Musman, M. Hotter, and J. Ostermann, "Object oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, vol. 1, pp. 117-138, October 1989.

[6] J. Osterman, "Object based analysis – synthesis coding based on the source model of moving rigid 3D objects," *IEEE Transactions on image processing*, vol. 3, pp. 705-711, September 1994.

[7] P. Gergen, "Object-based analysis-synthesis coding of image sequences at very low bit rates," *IEEE Transactions on Circuits and Systems for video technology*, vol. 4, No. 3, pp. 228-235, June 1994.

[8] ITU-T, *Video coding for low bit rate communication, draft recommendation - H.263, Study Group*, October 1995.

[9] P. L. Rosin, "Thresholding for change detection," *Department of Information Systems and Computing, Brunel University*, UK.

[10] P. K. Sahoo, "Threshold selection based on histogram modelling," *Department of Mathematics, University of Louisville*, Louisville, Kentucky, 40292.

[11] N. A. Tsoligkas, D. Xu, I. French and Y. Luo, "A motion model based video stabilisation algorithm," *World Automation Congress (WAC) 2006*, Budapest, Hungary, July 2006.

[12] S. Goss, W Vogt, R. M. Pelz, and D. Lappe, 'Transmission of still and Moving Images Over Narrowband Channels,' *Application Report*, Texas Instruments, SPRA110, February 1994.

[13] R. Talluri, "A hybrid Object – based video compression technique," *Proc. Int. Conf. Image Processing ICIP 96*, Lausanne, Switzerland, pp. 387-390, September 1996.

[14] O. Trier and A. Jain, "Goal–directed evaluation of binarization methods," *IEEE Trans. Pattern Anal. Machine Int.*, vol. 17, pp. 1191-1201, December 1995.

[15] P. Sahoo, S. Soltani, A. Wong, and Y. Chen, "A survey of thresholding techniques," *Compt. Vis. Graph. Image Processing*, vol. 41, pp. 233-260, 1988.

[16] A. Amer, "Memory based spatio temporal real time object segmentation," in Proc. SPIE, Conf. on Real-Time Imaging (RTI), Santa Clara, USA, vol. 5012, pp. 10-21, Jan. 2003.

[17] A. Amer, E. Dubois, "Image segmentation by robust binarization and fast morphological edge detection," In Proc. IAPR Conf. on Vision Interface pp. 357-364, Montréal, Canada, May 2000.

[18] K. Konstantinides, B. Natarajan, and G. Yovanof, "Noise estimation and filtering using block-based singular – value decomposition," *IEEE Trans. Image Processing*, vol. 6, pp. 479-483, March 1997.

[19] R. Mech and M. Wollborn, "A noise robust method for segmentation of moving objects in video sequences," *Universitat Hannover*, Institut fur Theoretische Nachrichtentechnik und Informationsverarbeitung.

[20] Y. Luo, D. Xu, I. French, and N. A. Tsoligkas, "A scheme for object-based video segmentation," *World Automation Congress (WAC) 2006*, Budapest, Hungary, July 2006.

[21] K. J. Kim, J. Y. Suh, D. Hyun Lee, C. W. Lim, and K. T. Park, "Shape Consistent Segmentation algorithm for extracting of moving object," *Proceedings of ICSP'96*, pp. 902-905, 1996.

[22] Y. Yong, H. Bo, W. Qiao, and W. Lenan, "An object–based Segmentation Approach for very low bit rate video sequences," *Proc. of 2001 International Symposium on intel. Multimedia, video and speech Processing*, Hong Kong, May 2001.

[23] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*, Kluwer Academic Publishers, 2003.

[24] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol.17, pp. 185-203, 1981.

[25] M. Ghanbari, "Video coding an Introduction to standard codecs," *IEE Telecommunications*, series 42, pp. 196-197, 1999.

[26] H. Schiller and M. Hotter, "Investigation on color coding in an object – oriented analysis-synthesis coder*," Signal Processing: Image Commun.,* vol. 5, pp. 319-326, October 1993.

[27] M. Wollborn, "Prototype prediction for color update in object – based analysis synthesis coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 236-245, June 1994.

[28] D. Mukherjee et al. "A region–based video coder using Edge Flow Segmentation and Hierarchical Affine region Matching" *Department of Electrical and Computer Engineering, University of California,* Santa Barbara, CA93106.

[29] K. J. Kim, J. Y. Suh, D. Hyun Lee, C. W. Lim, and K. T. Park, "Shape Consistent Segmentation algorithm for extracting of moving object," *Proceedings of ICSP'96*, pp. 902-905, 1996.