Motion Detection and Object Tracking in Video frame sequence on IWT of Adaptive Pixel Based Prediction Coding

Mr. K.Dinesh[1], Lecturer
Information Technology,
J.P College of college engineering,
Tirunelveli, India

Mrs. T. Arumuga Maria Devi[2], A.P
Centre for Information Technology,
Manonmaniam Sundaranar University,
Tirunelveli, India

Abstract:

To reduce the temporal, spatial and spectral redundancies in video frame sequence for the similarity between the current and the prior image, a new video coding algorithm based on integer wavelet transform with adaptive pixel-based prediction coding. A new scheme is presented which employs adaptive strategy based on Normalized cross correlation and partitioning for motion detection and object tracking in frame sequence. With the algorithm we have realized the real-time compression of video streams with high fidelity.

Keywords: Integer wavelet transforms (IWT), lossless video coding, pixel prediction, Normalized correlation and partitioning

1. Introduction:

Video compression is now essential for applications such as transmission and storage in data bases. According to the features of star-field video, LIU [1] exploited a zero-tree wavelet algorithm with extremely high compression. In fact, LIU's simple difference method is not always successful for there is more complicate content in most video sequence. The paper also improved LIU's motion detecting method using corner detection and matching algorithm [2]. Here, we use a motion detection and object tracking method called normalized cross correlation and partitioning to detect the object movement. And a backward pixel based predictive coding framework is presented which employs different strategy based on different video frames. Later, integer wavelet coding method is used to for still image compression. With the algorithm we can realize the real-time compression of video streams with high fidelity and high compression.

2. System framework:

The flow chart of coding is showed in Figure 1. The input of the system is the colour video frames captured by camera. The image is transformed from RGB colour space to YUV colour space, and then coded in YUV colour space. To improve the compression the weight of YUV is assigned as 4:2:2 considering the features of human vision. After conversion, Normalized cross correlation and partitioning is used for motion detection and object tracking. Then for each frame sequence integer wavelet transform is applied because motion compensation in the wavelet domain might be inefficient due to shift variant attributes of wavelet transform. Therefore, it might be unwise to process all kinds of video sequences in the spatial domain alone or in the wavelet domain alone. To avoid this, IWT is used. IWT uses mapping integer pixels to integer coefficients are used, which is suitable for lossless application. Later an pixel based predictive coding is applied to provide high compression.

The decoding process is adverse to the coding process.

3. Theory of NCC (Normalized Cross Correlation)

Manoj S. Nagmode et.al presented object detection step to have reliable, robust and fast visual surveillance system. The algorithm used gives better performance in terms of Detection Rate (DR) and processing time per frame.

Correlation is mainly used for measuring similarity between two images. It is useful in feature recognition and registration. Normalized cross correlation is given by equation (1).

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}} \quad (1)$$

In this, $\bar{A}$ and $\bar{B}$ indicates average pixel value in image A and B respectively. 'r' is normalized with

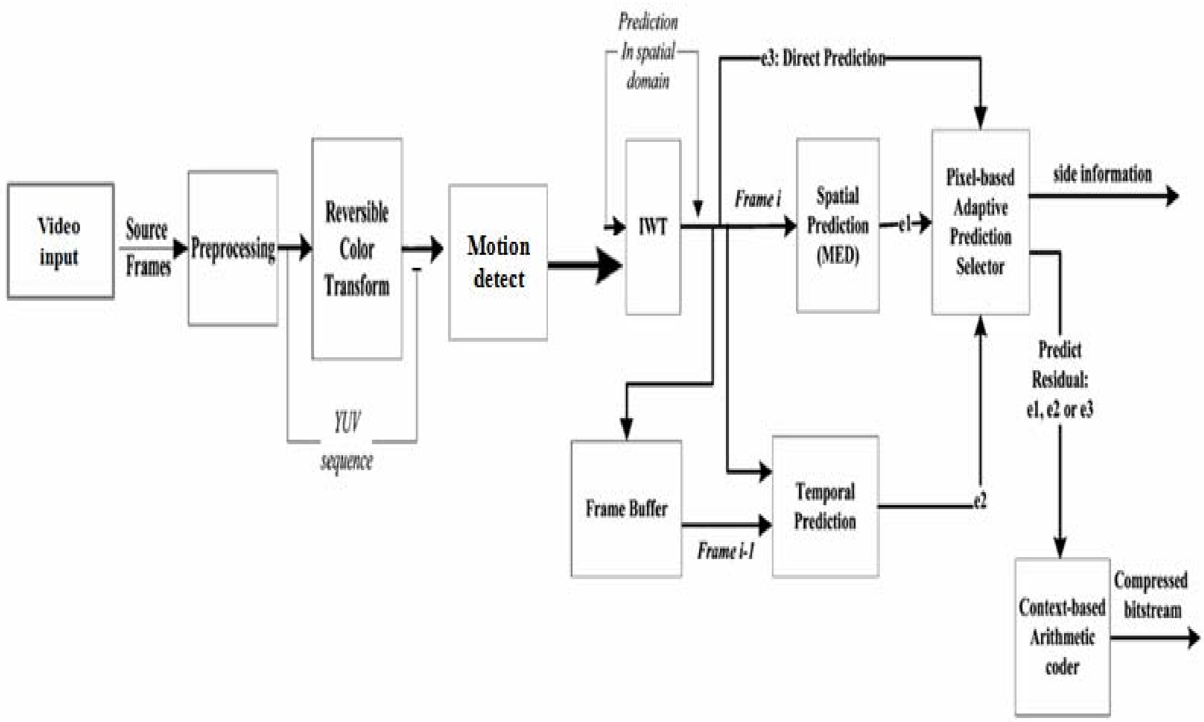


Figure 1. System Framework

respect to both the images and it always lies in the range [-1, 1].

4. System Overview

Basic steps involved in the process are given in figure 2. As shown, input image sequence is taken from the static camera. Two consecutive frames from the image sequence are partitioned into four quadrants. Then moving object detection takes place after finding Normalized Cross Correlation between two partitioned frames. Moving Object detection in video involves verifying the presence of an object in image sequence and possibly locating it precisely for recognition. After detecting the moving object, the location of the moving object is obtained by performing component is the connected analysis. Tracking of the detected moving object takes place by calculating the centroids of the detected moving object. Tracking means the detection of a target over time, thus establishing its trajectory. The aim of object tracking is to establish a correspondence between objects or object parts in consecutive frames and to extract temporal information about objects such as trajectory, posture, speed and direction.

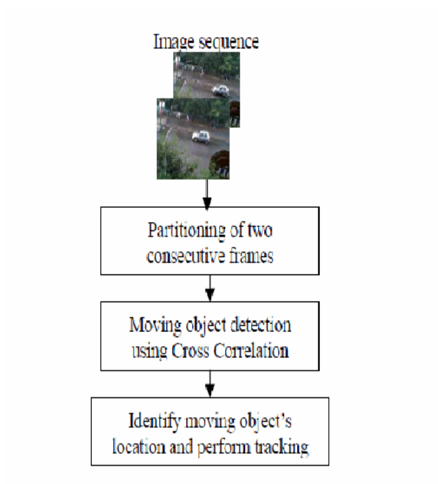


Figure 2. Basic steps

5. Algorithm

Basic algorithm steps for the detection and tracking of moving objects are given below.

• Read two consecutive frames from the image sequence called as current frame and previous frame
• Divide these frames into four quadrants.
• For ex: Current frame is divided into four parts called as x1, x2, x3 and x4. Similarly, previous frame is divided into four parts called as y1, y2, y3 and y4.
• Now find out the NCC of each sub image of current frame with the previous frame. After this there are four values of NCC, called as c1, c2, c3 and c4.
• Now find out the minimum value of NCC from these four values.
• To this minimum value of NCC apply the threshold.
• The threshold value is selected by taking average of four NCC values (i.e. c1, c2, c3 and c4).
• Suppose the minimum value of NCC is obtained at the first quadrant, it means that the moving object is present in that quadrant.
• Now operate in the first quadrant. Take the difference between the first quadrants of two consecutive frames.
• Then find the location of the moving object by performing component connected analysis and morphological processing.
• Centroid calculation is done for tracking the moving object.
• After this the second minimum value from the c1, c2, c3 and c4 is obtained. This is performed to check whether any other moving object is present in other part of the image.
• If the second minimum value is also greater than threshold then it means that the moving object is present in that quadrant. Now, identify the location of second moving object and track that object.
• Repeat the same procedure for the next frame.

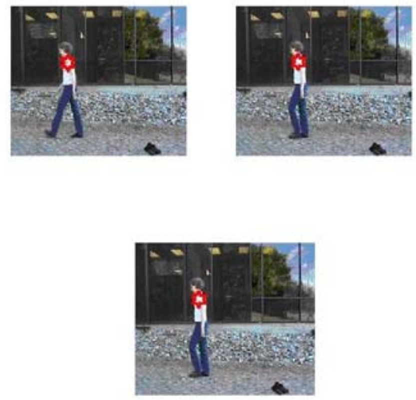


Figure 3 The tracking sequence of a walking person. This walking person is pointed by a red star

6. Adaptive pixel-based prediction coding

A. Pre-processing

Ying Li et.al presented a adaptive pixel-based prediction coding for video compression which gives better result. The wavelet transform has been extensively used for compression due to its excellent energy compaction characteristics. In the proposed scheme, an integer wavelet transforms (IWT) mapping integer pixels to integer coefficients is used, which is suitable for lossless application. Gong et al. [3] pointed out that motion compensation in the wavelet domain might be inefficient due to the shift-variant attributes of wavelet transform. Therefore, it might be unwise to process all kinds of video sequences in the spatial domain alone or in the wavelet domain alone. We present a method to determine the operational domain of a video sequence in two steps according to its temporal and spatial redundancies. In the first step, the amount of temporal redundancy is estimated by the interframe correlation coefficients of the test video sequence. For example, let us use $p_i(x,y)$ to represent the pixel to be encoded that is located at (x,y) in frame[i], then the interframe correlation coefficient between frame[i] and [i+1] can be calculated by **(1),** shown at the bottom of the page [4], where $\overline{pi}$ is the average value of the pixels in frame[i] ,$\overline{pi+1}$ is the average value of the pixels in frame[i+1]. If the average of the interframe correlation coefficients is smaller than a predefined threshold, then the sequence is likely to be a high motion video sequence. In this case, motion compensation in the wavelet domain would be inefficient; therefore, it is wise to operate on the sequence in the spatial domain. The threshold is set as 0.9 in experiments.

$$\rho_{i,i+1} = \frac{\sum_x \sum_y (p_i(x,y) - \overline{p_i}) \cdot (p_{i+1}(x,y) - \overline{p_{i+1}})}{\sqrt{\left(\sum_x \sum_y (p_i(x,y) - \overline{p_i})^2\right)\left(\sum_x \sum_y (p_{i+1}(x,y) - \overline{p_{i+1}})^2\right)}}$$

In the second step, we determine the suitable IWT for the test sequence by estimating its spatial redundancy. [3], [5] discussed that there is no one transform that has higher performance in lossless and lossy compression for all types of images as well as low computational complexity. Here, we use two common examples of IWT in the proposed scheme, the S and 5/3 transformations, based on the consideration of their simplicity and effectiveness, respectively. The S transform is the integer version of the Haar transform [5] which has the lowest computational complexity among all transforms, and the 5/3 transform performs reasonably well for both lossy and lossless compression [5]. The forward S transform equations are

$$d[n] = x[2n+1] - x[2n]$$
$$s[n] = x[2n] + \left\lfloor \frac{d[n]}{2} \right\rfloor .$$

The forward 5/3 transform equations are

$$d[n] = x[2n+1] - \left\lfloor \frac{x[2n] + x[2n+2]}{2} \right\rfloor$$
$$s[n] = x[2n] + \left\lfloor \frac{d[n-1] + d[n]}{4} + \frac{1}{2} \right\rfloor$$

Where x(n) is the input signal, d(n) is the high-frequency subband signal and s(n) is the low-frequency subband signal.

DCT coefficients are a approach used to estimate the amount of the spatial redundancy. Larger amplitudes from the high frequency coefficients imply that the adjacent pixels vary greatly within a frame, and a 5/3 wavelet transformation is used because it can exploit the spatial redundancy more effectively. Another similar approach used to approximate the intraframe redundancy is to observe the amplitude of the high frequency coefficients after IWT.

The proposed method using interframe and intraframe redundancy is suitable for RGB sequences, yet it does not always work for YUV sequences. For YUV video sequences, we find that operating in the spatial domain always provides the best compression performance. Sequence classification for the operational domain and the transformation type for a given video sequence is a hard task and the optimum approach remains an open question.

In the proposed algorithm, we use the above method to determine the operational domain of a video sequence if it is in the RGB color domain. The encoder will use a flag to represent the selected operational mode among spatial domain, transformation and 5/3 transformation. If a video sequence is in the YUV color domain, we will process it in the spatial domain without any transformation.

Once the operational domain of a video sequence is determined, it is transformed into the YUV color space if it is in the RGB color space to reduce color redundancy. This is implemented by applying the reversible color transform [6], [7] to all frames directly. For a video sequence to be processed in the wavelet domain, a one level IWT is applied to all frames after the color transform, and the encoder uses a flag to represent the selected transformation.

B. Spatial Prediction

To reduce the spatial redundancy, a prediction is computed based on the neighboring symbols in the same frame as the symbol to be encoded (here, we use the term symbol because it can be a pixel in the spatial domain or a wavelet coefficient in the wavelet domain). In the proposed scheme, we use a simple but robust spatial predictor, the median edge detector (MED), as used in JPEG-LS [8]. MED estimates the symbol to be encoded based on the values of the three previously encoded neighboring

Symbols. We use $p_i(x,y)$ to represent the symbol to be encoded that is located at (x,y) in frame[i]. The spatial predicted value of pi(x,y) is represented as

$$\hat{p_i}^S(x,y) = \begin{cases} \min(A,B), & \text{if } C \geq \max(A,B) \\ \max(A,B), & \text{if } C \leq \min(A,B) \\ A+B-C, & \text{otherwise} \end{cases}$$

Where

$$A = p_i(x-1,y), B = p_i(x,y-1), C = p_i(x-1,y-1)$$

Thus, the spatial prediction residual is

$$e_1 = p_i(x,y) - \hat{p_i}^S(x,y).$$

C. Temporal Prediction

Inspired by the temporal prediction scheme presented in [6], we introduce a novel adaptive pixel-based predictor based on the symbols in the reference frame with improvement to reduce temporal redundancy. The proposed temporal predictor is effective and accurate, and does not require the transmission of any extra side information.

Let pi(x,y) be the symbol to be encoded. The proposed temporal predictor aims to find the best matched symbol in reference frame [i-1]. Instead of exploiting the motion activity of pi(x,y) between adjacent frames directly, the predictor investigates the motion activity of the target window of pi(x,y) in frame[i] and frame[i-1] within a search range W*H where the target window is composed of the upper-left neighboring symbols of pi(x,y).

The temporal predictor of symbol pi(x,y).searches and locates the best matched target window in frame [i-1] which achieves the minimum cumulative absolute difference (CAD) within the search range, where

$$\mathrm{CAD}(T_w) = \sum_{(m,n)\in T_w} |p_i(x,y) - p_{i-1}(x+m, y+n)|$$

Where T_w denotes the target window, pi(x,y) and $p_{i-1}(x,y)$ denote the symbol values of the current frame [i] and the reference frame , respectively, and where a motion vector [m,n] is determined for the region , -W ≤ m≤ W, -H ≤ n ≤ H to minimize the CAD. Similar to block motion compensation techniques, the best motion vector for the target window with the minimum CAD is determined by

$$(m_0, n_0) = arg_{\{m,n\}} \{\min \mathrm{CAD}(T_w)\}$$

where (m_0,n_0) indicates the motion displacement of the target window. Then, the temporal predictor of can be obtained by

$$\hat{p_i}^T(x,y) = p_{i-1}(x+m_0, y+n_0)$$

and the temporal prediction residual is

$$e_2 = p_i(x,y) - \hat{p_i}^T(x,y).$$

Since the motion activity of the target window between frame [i] and frame [i-1] can be perfectly reconstructed at the decoder, there is no requirement for the transmission of the motion vector $[m_0 , n_0]$ to the decoder. Note that unlike [6] we are looking for a match of a window surrounding the symbol to be encoded rather than a block of symbols. As such, we expect better prediction accuracy.

Test results show that the above proposed temporal predictor achieves excellent performance. However, it needs to search and match the target windows in frame [i-1] for all symbols in frame[i] independently. This is very time consuming and places an undue computational burden in real applications.

In our implementation, this problem is solved by caching previous computation results. An important, though simple, observation is that the target windows of two neighboring pixels overlap significantly. Therefore, there is no need to repeat the computation for pixels in the overlapped portion. In addition, we propose a refined approach that can further greatly reduce the time for prediction, but might cause a slight decrease in compression ratio. Therefore, the refined approach should be used in applications that have higher temporal constraints.

The basic idea of the refined approach is that motion activities of neighboring symbols for a specific frame are different but highly correlated since they usually characterize very similar motion structures. Therefore, motion information of symbol pi(x,y). can be approximated by the motion information of the neighboring symbols in the same frame, and be refined over a relatively small search range with a relative small target window.

The initial motion vector (v_x ,v_y) of the current symbol pi(x,y). is approximated by the motion activity of the upper-left neighboring symbols in the same frame. It is the average motion vector of the four past neighboring symbols in (9), shown at the bottom of the page, where $(m_0 (x-1, y), n_0 (x-1,y))$ is the motion vector of symbol $p_i(x-1,y)$ which can be obtained by (10), and so on, for $(m_0 (x, y-1), n_0 (x,y-1))$, $(m_0 (x-1, y-1), n_0 (x-1,y-1))$ and $(m_0 (x+1, y-1), n_0 (x+1,y-1))$. The computation of (v_x,v_y) only uses the past information which is available at both the encoder side and the decoder side. The motion trajectory of symbol pi(x,y) is refined by finding the best match of a small target window $M_r * N_r$ within a small search range $W_r * H_r$.

The final motion estimation pi(x,y) of tightly relies on the initial motion vector (v_x ,v_y) ; thus, an unsound initial motion vector will have a negative impact on the final prediction performance. Another control parameter named pixel group, P_g is introduced in order to prevent that negative impact. Within each pixel group, the initial motion vector of each symbol is acquired according to (9). Then, it is refined in a smaller range. For symbols that have not enough neighbors to apply (9), the motion vectors can be acquired directly with the method discussed before, as shown in Fig. 5. When the number of encoded symbols achieves P_g, the procedure is re-initialized. The motivation for the refined scheme is to utilize the inherent connection of the motion information among neighboring symbols to significantly reduce the searching and matching time. Note that the compression efficiency is the first design objective of this paper. The refined approach only provide a potential solution to balance the tradeoff of compression efficiency and computation overhead. Therefore, all experiments in this work are conducted with the caching approach, which considerably reduces the searching and matching time. Considering that video data is no stationary, and the characteristics of different video sequence always vary greatly from each other. It is impossible to find a set of fixed parameter values that work well for all video sequences. Therefore, in the proposed algorithm, parameters M, N, L, W and H, are adjustable to improve the compression performance.

D. Direct Mode

If a video sequence is to be processed in the wavelet domain, then we use another prediction mode which is similar to the concept of direct sending mode as described in [6]. Because of the energy compaction property of the IWT, the wavelet coefficients in the high frequency sub bands (LH, HL, HH) usually have small amplitudes, which may be smaller than the amplitudes of the spatial prediction residuals and temporal prediction
residuals. Therefore, in this case the wavelet coefficients are encoded and transmitted directly denoted as .

E. Adaptive Backward Prediction Mode Selection

As stated before, our scheme contains two key points: one is the use of pixel based prediction (that is, the prediction is performed by pixel other than by block) to remove the temporal redundancy as much as possible as described in Section III-C, the other is the extremely low side information transmission which will be discussed in this section. This is accomplished by utilizing a simple but effective backward adaptive prediction mode selector. The scheme adaptively selects the predictor among three candidates (e_1 ,e_2 ,e_3) based on previous prediction accuracy.

The adaptive prediction selection is based on the sum of amplitudes of the prediction residuals of the past neighboring pixels as illustrated in Fig. 6. Suppose we want to determine the prediction mode of pi(x,y). We calculate

$$e^{(S)} = |e_1(x-1,y-2)| + |e_1(x,y-2)| + |e_1(x+1,y-2)| + |e_1(x-2,y-1)| + |e_1(x-1,y-1)| + |e_1(x,y-1)| + |e_1(x+1,y-1)| + |e_1(x-2,y)| + |e_1(x-1,y)|$$

which represents the sum of amplitudes of the spatial prediction residuals of the past neighboring symbols in Fig. 6, and

$$e^{(T)} = |e_2(x-1,y-2)| + |e_2(x,y-2)| + |e_2(x+1,y-2)| + |e_2(x-2,y-1)| + |e_2(x-1,y-1)| + |e_2(x,y-1)| + |e_2(x+1,y-1)| + |e_2(x-2,y)| + |e_2(x-1,y)|$$

which represents the sum of amplitudes of the temporal prediction residuals of the past neighboring symbols in Fig. 6, and

$$e^{(D)} = |e_3(x-1,y-2)| + |e_3(x,y-2)| + |e_3(x+1,y-2)| + |e_3(x-2,y-1)| + |e_3(x-1,y-1)| + |e_3(x,y-1)| + |e_3(x+1,y-1)| + |e_3(x-2,y)| + |e_3(x-1,y)|$$

which represents the sum of amplitudes of the direct prediction residuals of the past neighboring symbols in Fig. 6. The final prediction mode is indicated by the mode with the minimal value among $e^{(s)}$, $e^{(T)}$, $e^{(N)}$ which is

$$\text{mode} = \arg\left(\min_{\{S,T,N\}}\left\{e^{(S)}, e^{(T)}, e^{(N)}\right\}\right)$$

For example, if $e^{(T)}$ is the smallest value in { $e^{(s)}$, $e^{(T)}$, $e^{(N)}$ } for symbol pi(i ,j) , then temporal prediction is selected as the prediction mode and the prediction residual is e_2 as discussed in Section III-C. The selection and calculation of the prediction mode only uses the past information; hence, it has the advantage of not requiring the transmission of any extra side information. This approach not only reduces the size of the compressed data by removing the extra bits used to represent the prediction mode, but also achieves high prediction efficiency by adaptively selecting the predictor that performs best for neighboring pixels. For an entire video sequence, the side information needed to be transmitted includes the following items:

- transformation type (, 5/3 or spatial); this is only applied to the RGB video sequences;
- M, N and L specify the size of the used target window;

- W, H specify the size of the used search range;
- frame width and height;
- original color space (RGB or YUV).

As we can see from the above list, the spatial requirement for the side information is O(1) . That is, the size of the side information is irrelevant to the size of video sequences.

F. Context Modeling

Context modeling is used for efficient coding of the prediction residuals. By utilizing suitable context models, the given prediction residual can be encoded by switching between different probability models according to already encoded neighboring symbols of the symbol to be encoded.

In the proposed scheme, two causal context models are used. One is used for intraframe symbols and another is used for interframe symbols as illustrated in Fig. 7. In the intraframe mode, a nine-symbol context is built for the current symbol pi(x,y) In the interframe mode, temporal redundancy is exploited by using symbols from the same neighborhood forming a corresponding block in frame[i-1] as shown in Fig. 7, and a nine-symbol context is built for symbol pi(x,y) The prediction residuals under the selected mode are encoded with one of the corresponding context models. The context template can be obtained by

$$C = |e_0| + |e_1| + |e_2| + |e_3| + |e_4| + |e_5| + |e_6| + |e_7| + |e_8|$$

Where e_i is the prediction residual of the corresponding symbol p_i as shown in Fig. 7. Each context mode is quantized into 16 regions, where the quantizers

$$q_1 = 1, q_2 = 3, q_3 = 5, q_4 = 7$$
$$q_5 = 9, q_6 = 11, q_7 = 13, q_8 = 15,$$
$$q_9 = 17, q_{10} = 21, q_{11} = 25, q_{12} = 32,$$
$$q_{13} = 44, q_{14} = 70, \text{ and } q_{15} = 96,$$

respectively. The quantizers are determined by observing the histogram of the prediction residuals experimentally. It can be optimized in different applications. The prediction residuals are encoded with the corresponding context followed by an adaptive arithmetic coder as discussed in [11].

7. Conclusion

We presented a new scheme to reduce the temporal, spatial and spectral redundancies in video frame sequence. A new scheme is presented which employs adaptive strategy based on Normalized cross correlation and also realized that the real-time compression of video streams with high fidelity to detect and track the object in frame sequence.

References

1. LIU Weifeng and WANG Zengfu, A high compression algorithm for stereo video stream of star field,ICSIT'2005, November 2005

2. Zhou Peng, Tan Yong, Xu Shoushi, A New Method of Image Registration Based on Corner Detection, Journal of University of Science and Technology of China, vol.32, No.4, Oct. 2002

3. Y. Gong, S. Pullalarevu, and S. Sheikh, "A wavelet-based lossless video coding scheme," in Proc. Int. Conf. Signal Processing, 2004, pp. 1123–1126.

4. M. G. Bulmer, Principles of Statics. New York: Dover, 1979.

5. A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, "Wavelet transforms that map integers to integers," Appl. Comput. Harmon. Anal., vol. 5, no. 3, pp. 332–369, Jul. 1998.

6. S.-G. Park, E. J. Delp, and H. Yu, "Adaptive lossless video compression using an integer wavelet transform," in Proc. Int. Conf. Image Processing, 2004, pp. 2251–2254.

7. D. S. Taubman and M. W. Marcellin, JPEG2000 Image Compression Fundamentals, Standards and Pratice. Norwell, MA: Kluwer, 2002.

8. JPEG-LS, Lossless and Near-Lossless Coding of Continuous Tone Still Images, ISO/IEC JTC1/SC 29/WG 1, Jul. 1997.

9. S.-G. Park, "Adaptive lossless video compression," Ph.D. dissertation, School Elect. Comput. Eng., Purdue Univ., West Lafayette, IN, Dec. 2003.

10. Z. Ming-Feng, H. Jia, and Z. Li-Ming, "Lossless video compression using comnination of temporal and spatial prediction," in Proc. IEEE. Int. Conf. Neural Netwoks Signal Processing, Dec. 2003, pp. 1193–1196.

11. K. Sayood, Introduction to Data Compression, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2000.

12. N. D. Memon and K. Sayood, "Lossless compression of video sequences," IEEE Trans.

Commun., vol. 44, no. 10, pp. 1340–1345, Oct. 1996.

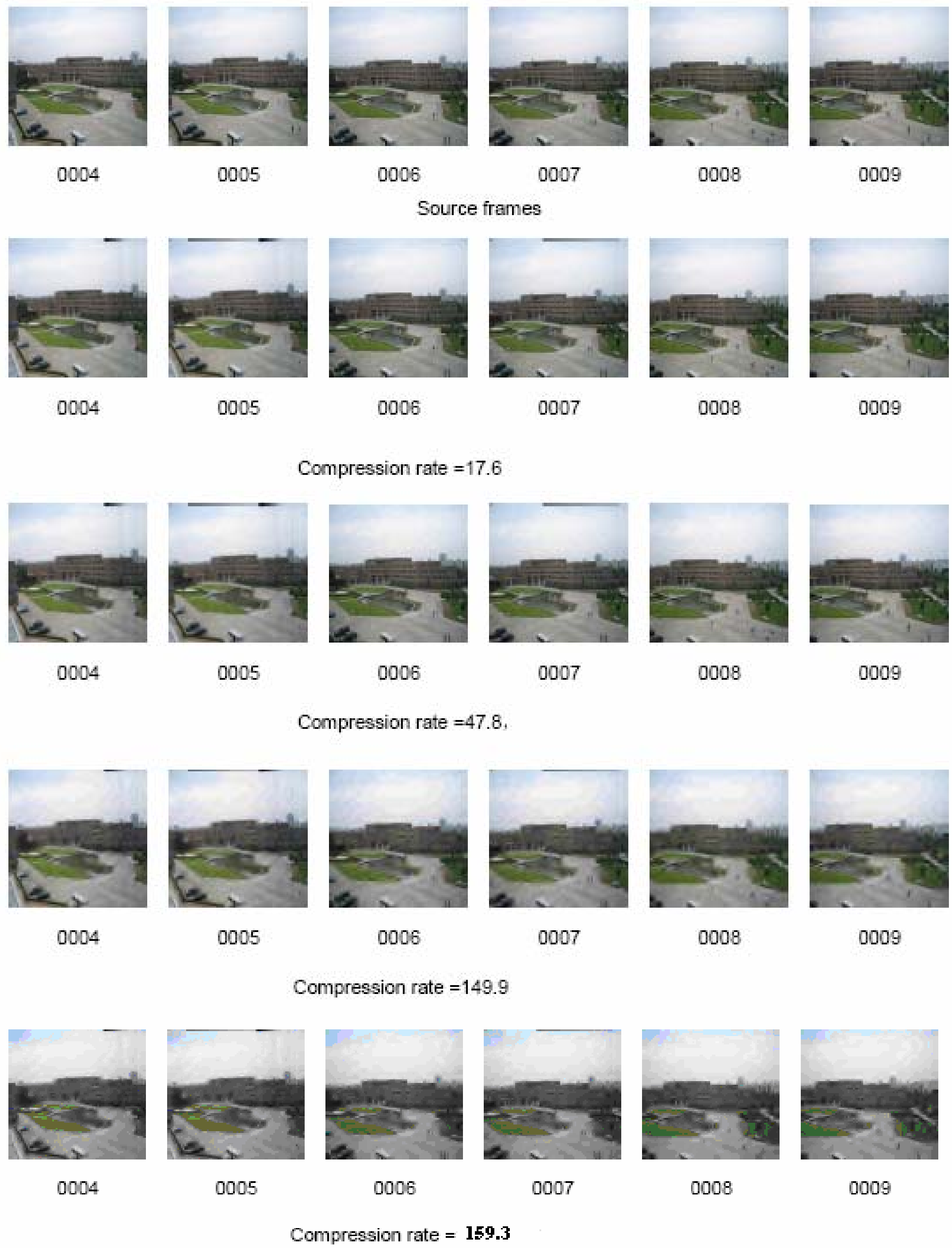
13. K. H. Yang and A. F. Faryar, "A context-based predictive coder for lossless and near-lossless compression of video," in Proc. Int. Conf. Image Processing, Sep. 2000, vol. 1, pp. 144–147.

14. E. S. G. Carotti, J. C. De Martin, and A. R. Meo, "Low-complexity lossless video coding via spatio-temporal prediction," in Proc. Int. Conf. Image Processing, Sep. 2003, vol. 2, pp. 197–200.

15."Backward-adaptive lossless compression of video sequences," in Proc. IEEE Int. Conf. Audio, Speech, Signal Processing, Apr. 2002, pp. 3417–3420.

K.Dinesh received B.Tech Degree in Information Technology and Management from SASTRA University, Thanjore, India in 2009, He completed his M.Tech degree in Computer and Information Technology at Manonmaniam sundaranar University. Currently he is working as a lecturer in J.P College of engineering. His research interests include Signal and Image Processing.

T. Arumuga Maria Devi received B.E. Degree in Electronic and Communication Engineering from Manonmaniam Sundaranar University, Tirunelveli India in 2003, M.Tech degree in Computer and Information Technology from Manonmaniam Sundaranar University, Tirunelveli, India in 2005. Currently, she submitted Ph.D Thesis in Information Technology – Computer Science Engineering and also the Assistant Professor of Centre for Information Technology and Engineering of Manonmaniam Sundaranar University. Her research interests include Signal and Image Processing, Multimedia and Remote Communication.



Some experimental results