Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp195-200)

Temporal Video Compression By Discrete Wavelet Transform

L.P. TEO[1], W.K. LIM[2], W.N. TAN[2], Y.F. TAN[2], H.T. TENG[2], Y.F. CHANG[2]
[1] Faculty of Information Technology,
[2] Faculty of Engineering,
Multimedia University, 63100 Cyberjaya, Selangor, Malaysia.
http://www.mmu.edu.my

Abstract: This paper investigates the use of discrete wavelet transform in temporal video compression. The proposed method has the advantage of being independent of the number of frames. After decompression, it also reproduces video images that have relatively high quality of visualization.

Keywords: temporal video compression, discrete wavelet transform, Daubechies wavelet.

1. Introduction

In the world of information technology, the transmission of multimedia data is inevitably an enormous challenge. The most obvious and direct way to accommodate this demand is to increase the bandwidth available to all users. But the ever growing massive traffic, caused by the inherent nature of multimedia data, is not supported well by the existing technologies of transmission. Another solution to this problem is to compress multimedia data before storage and transmission, and decompress it at the receiver [1].

The fundamental of video compression research is to seek a method or algorithm to compress by extracting only the visible element and eliminating those redundant, thus substantially reducing the data needed to be stored, transmitted or used. The main objective of video compression is to reduce the data size for transmission or storage whilst maintaining an acceptable image quality. Among the most recent compression algorithms, the block matching algorithm [2, 3] provides a high compression ratio and is used in many popular video coding standards, namely H.261, MPEG-1 and MPEG-2. The basic idea for block matching in reducing the file size is by analyzing the video frames to determine the vector for motion region from frame to frame. Thus instead of transmitting the images, the initial image and the motion vectors are transmitted with a much smaller data size.

Recently, image and video compression gain greatly by the increasing usage of wavelet-based technologies. The characteristics of wavelets are especially compatible to low bit rate coding. Compression methods based on wavelets are widely acknowledged as producing results superior to traditional block-based compression schemes [4] such as JPEG.

In this paper, we discuss compression of a set of sequential images or movie video. Our idea is to compress the data by treating the intensity of each pixel for sequential images as a temporal or time element. The data, i.e., the intensity values of the pixel in the time domain are transformed using the Daubechies wavelet transform [5, 6, 7]. The resulting data has a lot of values that are closed to zero. We apply a threshold to cut off. As a result, the data stored for decompression contains a high percentage of zeros. The decompression is done by using the inverse wavelet transform applied to the compressed data.

This paper is parallel to our previous work where we considered temporal video compression by polynomial fitting [8].

2. Temporal compression and decompression

The discrete wavelet transform (DWT) has been widely used for still image compression [9, 10, 11]. Here we propose to use it for temporal video compression. Nevertheless, the basic idea is the same. We consider a sequence of video images of size $m \times n$ and label them by $t = 1, 2, \ldots, T$. We denote by $c_{ij}(t)$ the intensity value of the pixel at position (i,j) of the t-th frame. For each (i,j), we apply DWT to the sequence $\{c_{ij}(t)\}$ and obtain the detail coefficients sequence $\{d_{ij}(1,t)\}$ and the approximation coefficients sequence $\{a_{ij}(1,t)\}$. Then we apply DWT again to the detail coefficients sequence $\{d_{ij}(1,t)\}$ and obtain the sequences $\{d_{ij}(2,t)\}$ and $\{a_{ij}(2,t)\}$. This procedure is repeated for a fixed number of times, say N. The data that are needed for recovering the original sequence $\{c_{ij}(t)\}$ are all the approximation coefficients sequences $\{a_{ij}(1,t)\},\ldots,\{a_{ij}(N,t)\}$ and the detail coefficients sequence $\{d_{ij}(N,t)\}$. However, the approximation coefficients sequences contain a lot of values that are relatively small, especially if the pixel does not involve large movements. To achieve the compression purpose, we fix thresholds $V_1,\ldots,V_N$ and U for the sets $\{a_{ij}(1,t)\},\ldots,\{a_{ij}(N,t)\}$ and $\{d_{ij}(N,t)\}$ respectively and set those values whose absolute values are less than the thresholds to be zero. The resulting sequences $\{A_{ij}(1,t)\},\ldots,\{A_{ij}(N,t)\}$ and $\{D_{ij}(N,t)\}$, which contain a lot of zeros, are ideal for applying lossless compression techniques before being stored for decompression. To decompress, we simply apply the inverse DWT to the stored data.

In this paper, the wavelet we are using is the Daubechies wavelet db2. The reason we choose this wavelet is that it is the simplest continuous wavelet. This wavelet is determined by four coefficients p_1, p_2, p_3, p_4, where

$$p_1=\frac{1+\sqrt{3}}{4\sqrt{2}},\ p_2=\frac{3+\sqrt{3}}{4\sqrt{2}},\ p_3=\frac{3-\sqrt{3}}{4\sqrt{2}},\ p_4=\frac{1-\sqrt{3}}{4\sqrt{2}}.$$

The DWT is given by

$$d_{ij}(1,t)=p_1c_{ij}(2t-1)+p_2c_{ij}(2t)+p_3c_{ij}(2t+1)+p_4c_{ij}(2t+2),$$

$$a_{ij}(1,t)=p_4c_{ij}(2t-1)-p_3c_{ij}(2t)+p_2c_{ij}(2t+1)-p_1c_{ij}(2t+2);$$

While the inverse DWT is given by

$$C_{ij}(2t-1)=p_1D_{ij}(1,t)+p_3D_{ij}(1,t-1)+p_4A_{ij}(1,t)+p_2A_{ij}(1,t-1),$$

$$C_{ij}(2t)=p_2D_{ij}(1,t)+p_4D_{ij}(1,t-1)-p_3A_{ij}(1,t)-p_1A_{ij}(1,t-1).$$

Hence in order to recover the coefficients $C_{ij}(t)$ for t=1 to T, we need the coefficients $D_{ij}(1,t)$ and $A_{ij}(1,t)$ for t=0 to $T/2$ or $(T+1)/2$ depending on T is even or odd, and to recover $D_{ij}(1,t)$ for $t=0$ to $\lceil T/2\rceil$, we need $D_{ij}(2,t)$ and $A_{ij}(2,t)$ for $t=-1$ to $\lceil T/4\rceil$, where $\lceil * \rceil$ is the ceiling function. Iteratively, we need to store the values $\{A_{ij}(1,t), t=0,1,\ldots,\lceil T/2\rceil\}$,…, $\{A_{ij}(N,t), t=-1,0,1,\ldots,\lceil T/2^N\rceil\}$ and $\{D_{ij}(N,t), t=-1,0,1,\ldots,\lceil T/2^N\rceil\}$. On the other hand, to obtain $A_{ij}(N,t)$ and $D_{ij}(N,t)$ from $t=-1$ to K, we need to have $d_{ij}(N-1,t)$ for $t=-3$ to $2K+2$. Iteratively, we need the data $c_{ij}(t)$ for $t=-2^{N+1}+1$ to $2^N K+2^{N+1}-2$, where $K=\lceil T/2^N\rceil$. Hence we need to define the values of $c_{ij}(t)$ for $t\le 0$ and $t\ge T$. To guarantee the smoothness of data, we first extend $c_{ij}(t)$ so that it is symmetric with respect to the line $t=T$ and then extend it periodically so that it has period $2T-2$. Note that for fixed N, the number of additional frames we need is at least $2(2^{N+1}-2)$, and at most $2(2^{N+1}-2)+2^N-1$. It depends only on N and not on T. For T large (which is usually the case in applications), the number of additional frames becomes negligible.

3. Result and discussion

We use two sets of video files. The first one shows Claire announcing news. This set has T=50 frames and each frame contains 176×144 pixels. The second one shows Wee-Keong leaving his office. This set has T=60 frames and each frame contains 320×240 pixels. There are relatively small movements in the first set, whereas the movements of Wee-Keong are relatively large in the second set. We apply the Daubechies wavelet db2 to perform the DWT for $N=4$ and $N=5$ times. We then set the values that are within a certain threshold to zero. The resulting data is saved for decompression. To compare the decompressed data with the original data, we calculate the peak signal to noise ratio (PSNR) of each frame. In Tables 1 and Table 2, we show the percentage of zeros of the stored data and the average of the PSNR values using the thresholds $V_1=\cdots=V_N=U=Th$ by varying Th. In Figures 1, 2, 3 and 4, we depict the relations graphically.

The values of $c_{ij}(t)$ lie in the interval [0,1]. From the formulas in Section 2, it is easy to see that the values of $d_{ij}(1,t)$ lie in the interval $\left[p_4,\sqrt{2}-p_4\right]$. Since p_4 is a relatively small number, we can approximate it by zero. Therefore the values of $d_{ij}(k,t)$ lies in an interval that is slightly larger than the interval $\left[0,\sqrt{2^k}\right]$. Hence the uniform threshold we apply above is less significant for the coefficients in the higher level of N. In order to circumvent this problem, we can instead apply a normalized threshold.

Th	*N*=4		*N*=5	
	Percentage of zeros (%)	Average PSNR	Percentage of zeros (%)	Average PSNR
0.050	90.05	197.9917	91.58	195.7828
0.055	90.19	201.6620	91.75	196.0149
0.060	90.31	201.6748	91.89	195.2678
0.065	90.42	201.3556	92.02	195.1797
0.070	90.51	201.0721	92.13	195.2213
0.075	90.59	200.7969	92.23	196.3218
0.080	90.66	200.5366	92.32	196.7969
0.085	90.73	200.1821	92.40	199.6787
0.090	90.78	199.9130	92.47	199.4551
0.095	90.84	199.2491	92.54	199.2095
0.100	90.89	198.9628	92.60	198.8220
0.105	90.93	198.7349	92.66	198.5505
0.110	90.98	198.4428	92.72	198.2635
0.115	91.02	198.1862	92.77	197.9963
0.120	91.06	197.9400	92.82	197.7431
0.125	91.09	197.5855	92.86	197.4498
0.130	91.13	197.3587	92.90	197.1806
0.135	91.15	197.1545	92.94	196.9531
0.140	91.18	196.7293	92.97	196.5895
0.145	91.21	196.5549	93.01	196.3080
0.150	91.23	196.2868	93.04	196.2751
0.155	91.26	196.0962	93.07	196.0163
0.160	91.28	195.8801	93.10	195.7876
0.165	91.30	195.6527	93.12	195.6860
0.170	91.32	195.4593	93.15	195.4129
0.175	91.41	194.6711	93.18	195.1458
0.180	91.43	194.5217	93.20	194.9391
0.185	91.44	194.3633	93.22	194.6797
0.190	91.47	194.1197	93.24	194.4022
0.195	91.48	193.9803	93.26	194.2377
0.200	91.49	193.7770	93.28	193.9342

Table 1: The percentage of zeros of the stored data and comparison using PSNR of the compressed video showing Claire announcing news.

Th	*N*=4		*N*=5	
	Percentage of zeros (%)	Average PSNR	Percentage of zeros (%)	Average PSNR
0.050	79.40	189.6155	79.57	189.4340
0.055	80.44	188.2430	80.82	187.9859
0.060	81.35	187.0019	81.84	186.6131
0.065	82.14	185.7470	82.70	185.3094
0.070	82.85	184.5211	83.46	184.2666
0.075	83.48	183.5689	84.13	183.0189
0.080	84.06	182.3835	84.73	182.1592
0.085	84.58	181.4918	85.28	181.1953
0.090	85.06	180.5711	85.78	180.1317
0.095	85.50	179.6478	86.23	179.2841
0.100	85.91	178.9114	86.66	178.5773
0.105	86.28	178.6012	87.06	178.2091
0.110	86.63	177.7588	87.43	177.3087
0.115	86.95	176.9577	87.77	176.5395
0.120	87.25	176.3352	88.10	176.0012
0.125	87.53	175.7669	88.40	175.5305
0.130	87.79	175.2824	88.69	175.0667
0.135	88.03	174.5372	88.96	174.4607
0.140	88.26	174.0576	89.22	173.9633
0.145	88.47	173.9023	89.46	173.6970
0.150	88.66	173.4980	89.68	173.2398
0.155	88.85	173.1647	89.89	172.9896
0.160	89.03	172.7200	90.09	172.4902
0.165	89.20	172.4275	90.28	172.2328
0.170	89.37	172.1551	90.46	171.7822
0.175	89.52	171.8745	90.63	171.4942
0.180	89.67	171.3580	90.80	171.0654
0.185	89.81	170.8411	90.96	170.6258
0.190	89.94	170.3917	91.11	170.1196
0.195	90.06	169.7629	91.25	169.5560
0.200	90.19	169.4427	91.39	169.1871

Table 2: The percentage of zeros of the stored data and comparison using PSNR of the compressed video showing Wee-Keong leaving his office.

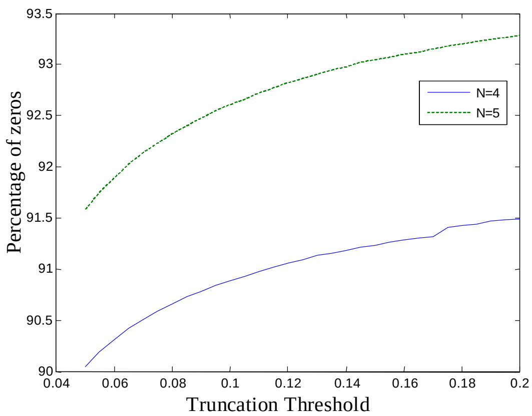


Fig. 1: The percentage of zeros of the stored data of the video showing Claire announcing news.

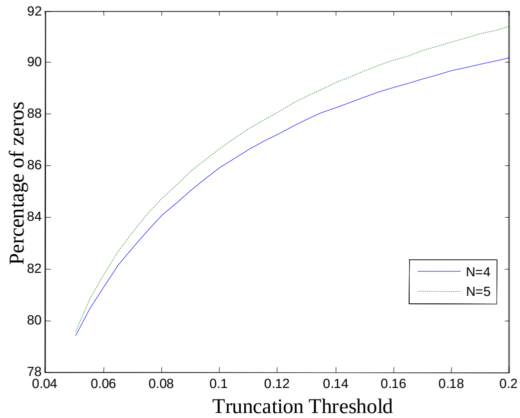


Fig. 2: The percentage of zeros of the stored data of the video showing Wee-Keong leaving office.

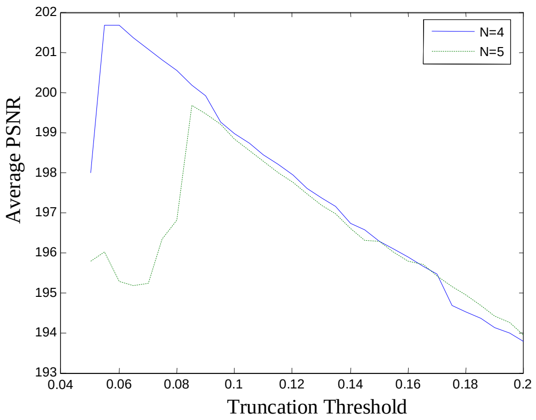


Fig. 3: The average PSNR of the compressed video showing Claire announcing news.

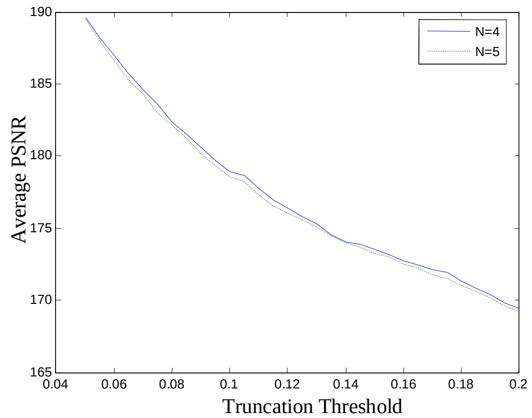


Fig. 4: The average PSNR of the compressed video showing Wee-Keong leaving office.

In Table 3 and Table 4, we show the percentage of zeros of the stored data and the average of the PSNR values using the normalized thresholds

$$V_k = \left(\sqrt{2}\right)^k Th \text{ and } U = V_N \text{, for } k = 1, 2, \ldots, N.$$

In Figures 5, 6, 7 and 8, we depict the resulted relations graphically.

From the tables and figures, we can see that the percentage of zeros increases when we increase the truncation threshold, as it should be the case. For video with less movement, we can achieve a percentage of more than 90% even with a mild threshold. For video with larger movement, we need a larger threshold, but still below 0.1 or 10% of the data value. Comparing the results of $N=4$ and $N=5$, we find that $N=4$ in general gives poorer percentage of zeros but higher average PSNR. However, for the same threshold value, the increase in the percentage of zeros is relatively larger than the decrease in PSNR when we go from N=4 to N=5. This means that higher N value is preferable.

Figures 9 and 10 show the 10th, 20th, 30th, 40th frames of the original images and the decompressed images using N=5 and normalized thresholds 0.02, 0.04, 0.06 and 0.08 respectively. Visually, the images showing Claire announcing news is not affected that much when we increase the threshold. However, the images showing Wee-Keong leaving office become a little blur in the part when large movements are involved.

The number of frames we used is small compared to actual application. Therefore we cannot ignore the fact that in order to decompress, the number of frames involved in compression is almost twice the number of decompressed frames. This considerably decreases the percentage of zeros of the stored data. We have tried the same algorithm with larger number of total frames, with the same images repeated by reflection and periodicity. The result reveals a higher percentage of zeros when all other parameters are fixed.

4. Conclusion and suggestion

The DWT has a great potential for temporal video compression. It has an advantage that it is independent of the number of frames involved. Moreover, there are a lot of parameters we can adjust which will not affect much the complexity of compression and decompression. In general, we can choose different truncation thresholds for different pixels and even for different part of the same pixel, according to how much a particular pixel value change with respect to time. In this way, we can expect to obtain decompressed image of more uniform quality while maintaining a high compression rate.

This research is still in the preliminary stage and there are still a lot of things to be explored. For future work, we will like to adjust the thresholds according to the range of values of a particular pixel. On the other hand, we will also like to explore the effect of combining the temporal compression and spatial compression using DWT. Another potential direction is to use the discrete wavelet packet transform, which we expect will give better compression rate than DWT.

Th	*N*=4		*N*=5	
	Percentage of zeros (%)	Average PSNR	Percentage of zeros (%)	Average PSNR
0.020	90.10	201.8320	91.93	200.7233
0.025	90.47	200.3005	92.35	199.5898
0.030	90.73	199.3306	92.66	198.0848
0.035	90.93	198.0571	92.88	196.5914
0.040	91.07	197.0649	93.05	195.5698
0.045	91.26	195.6764	93.23	193.7228
0.050	91.35	194.7995	93.34	192.3245
0.055	91.44	193.1867	93.44	191.3973
0.060	92.31	186.5364	94.12	184.8721
0.065	92.36	186.3015	94.18	184.4858
0.070	92.40	185.2483	94.23	183.9514
0.075	92.44	184.4705	94.27	183.6305
0.080	92.49	184.0766	94.32	183.1304
0.085	92.53	183.7024	94.36	182.7291
0.090	92.57	183.4431	94.40	181.7854
0.095	92.61	182.7185	94.44	181.3169
0.100	92.66	181.8736	94.48	180.8692

Table 3: The percentage of zeros of the stored data truncated by normalized thresholds and comparison using PSNR of the video showing Claire announcing news

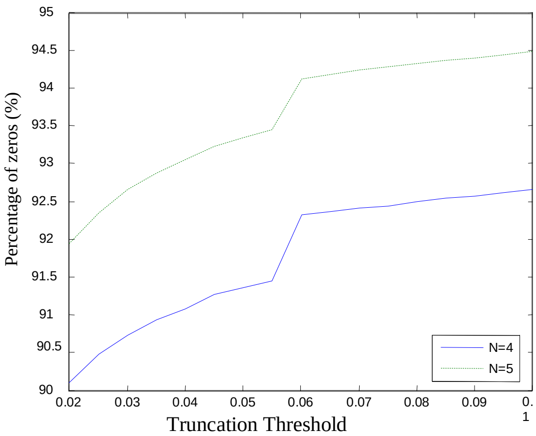


Fig. 5: The percentage of zeros of the stored data of the video showing Claire announcing news (normalized threshold).

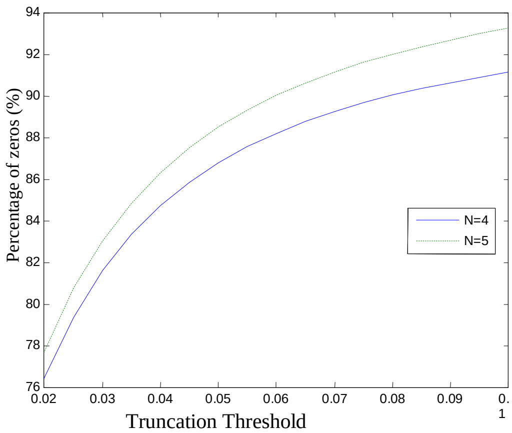


Fig. 6: The percentage of zeros of the stored data of the video showing Wee-Keong leaving office (normalized threshold).

Th	*N*=4		*N*=5	
	Percentage of zeros (%)	Average PSNR	Percentage of zeros (%)	Average PSNR
0.020	76.39	188.5671	77.67	187.1450
0.025	79.36	184.8766	80.74	182.9783
0.030	81.60	181.8875	83.05	180.0783
0.035	83.33	179.0942	84.84	177.2356
0.040	84.71	177.5060	86.29	175.2974
0.045	85.84	175.5307	87.49	173.4593
0.050	86.76	174.0096	88.48	171.6009
0.055	87.54	172.4864	89.31	170.3559
0.060	88.19	171.4356	90.01	169.0045
0.065	88.75	170.3401	90.62	167.9949
0.070	89.23	169.1557	91.14	167.0199
0.075	89.65	168.1464	91.60	166.7494
0.080	90.02	167.2648	91.99	165.6349
0.085	90.34	166.7068	92.35	164.9125
0.090	90.63	166.0212	92.68	163.7367
0.095	90.89	165.1545	92.97	162.6635
0.100	91.12	164.4932	93.23	162.0658

Table 4: The percentage of zeros of the stored data truncated by normalized thresholds and comparison using PSNR of the video showing Wee-Keong leaving office.

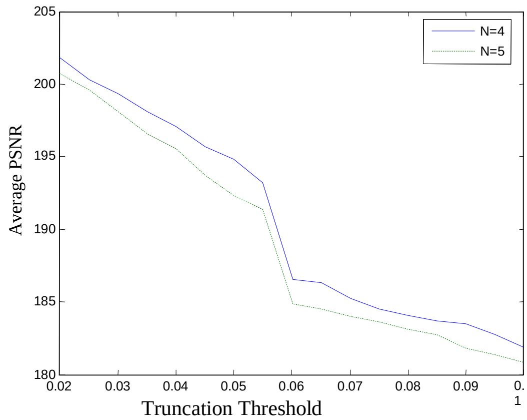


Fig. 7: The average PSNR of the compressed video showing Claire announcing news (normalized threshold).

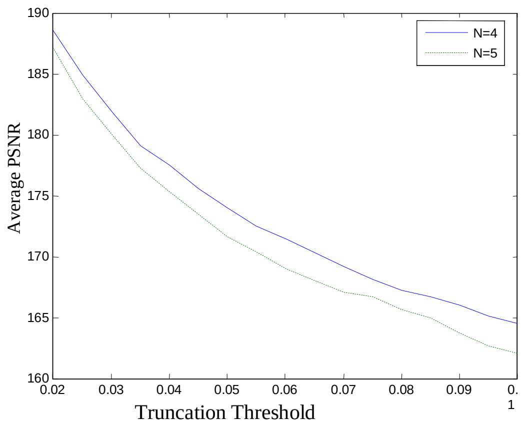
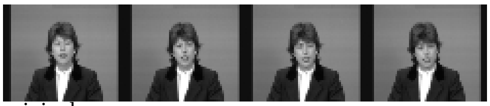


Fig. 8: The average PSNR of the compressed video showing Wee-Keong leaving office (normalized threshold).


original

normalized threshold=0.02

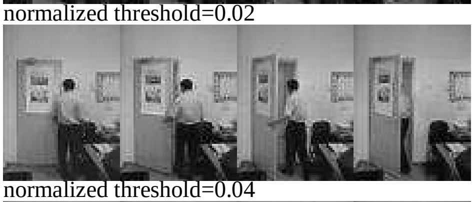
normalized threshold=0.04

normalized threshold=0.06

normalized threshold=0.08

Fig. 9: The 10th, 20th, 30th, 40th frames of the original and decompressed images of Claire announcing news using N=5 and normalized threshold 0.02, 0.04, 0.06 and 0.08 respectively.


original

normalized threshold=0.02

normalized threshold=0.04

normalized threshold=0.06

normalized threshold=0.08

Fig. 10: The 10th, 20th, 30th, 40th frames of the original and decompressed images of Wee-Keong leaving office using N=5 and normalized threshold 0.02, 0.04, 0.06 and 0.08 respectively.

Acknowledgment:
This project is partially supported by MMU internal funding (Project No: PR/2006/0584).

References:

[1] Symes, P.D., *Digital Video Compression*, McGraw Hill, 2003.

[2] Tziritas, G., Labit, C., *Motion Analysis for Image Sequence Coding*, Elsevier, 1994.

[3] Li, H., Lundmark, A. and Forchheimer, R., Image Sequence Coding At Very Low Bitrates: A Review, *IEEE Trans, Image Processing*, vol. 3, No. 5, 1994, pp. 589-609.

[4] E. Moyano, E., Orozco-Barbosa, L, , Quiles, F.J. and Garrido, A., A New Fast 3D Wavelet Transform Algorithm for Video Compression, *Workshop on Digital and Computational Video*, 2001, pp. 118-125.

[5] Koornwinder, T.H., *Wavelets: An Elementary Treatment of Theory and Applications*, Singapore: World Scientific, 1993.

[6] Walker, J.S., *A Primer on Wavelets and Their Scientific Applications*, New York: Chapman & Hall/CRC, 1999.

[7] Rebertson, M.A., Temporal Filtering of Wavelet-Compressed Motion Imagery, *International Conference on Image Processing (ICIP)*, 2004, pp. 295-298.

[8] Chang, Y.F., Lim, W.K., Tan, W.N., Tan, Y.F., Teng, H.T., Teo, L.P., Temporal Video Compression By Polynomial Fitting, *Proceedings of MMU Int. Symposium on Information & Communications Technologies (M2USIC)*, 2005, pp. TS03-9 – TS03-12.

[9] DeVore, R.A., Jawerth, B., and Lucier, B.J., Image Compression Through Wavelet Transform Coding, *IEEE Trans. on Information Theory*, Vol. 38, Issue 2, Part 2, 1992, pp. 719 – 746.

[10] Wang, J. and Huang, K., Medical Image Compression By Using Three-Dimensional Wavelet Transformation, *IEEE Trans. on Medical Imaging*, Vol. 15, Issue 4, 1996, pp. 547 – 554.

[11] Averbuch, A., Lazar, D. and Israeli, M., Image Compression Using Wavelet Transform And Multiresolution Decomposition, *IEEE Trans. on Image Processing*, Vol. 5, Issue 1, 1996, pp. 4–15.