Low Power Motion Estimation Design Using Adaptive Pixel Truncation

Zhong-Li He, Kai-Keung Chan, Chi-Ying Tsui, and Ming L. Liou
Dept. of Electrical and Electronic Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract

Power consumption is very critical for portable video applications such as portable video-phone. Motion estimation in the video encoder requires huge amount of computation and hence consumes the largest portion of power. In this paper we propose a novel method of reducing power consumption of the motion estimation by adaptively changing the pixel resolution during the computation of the motion vector. The pixel resolution is changed by masking or truncating the LSBs of the pixel data which is governed by an adaptive mechanism. Experimental results show that on average more than 4 bits can be truncated without affecting the picture quality. This results in an average 70% reduction in power consumption.

I. Introduction

As the multi-media and wireless technologies become mature, more and more sophisticated portable multimedia applications, such as wireless video phone and hand-held digital video camcorder, are becoming available. Real time video compression is required to reduce the bandwidth either for transmission or for storage of the video data. However, it consumes a lot of power. Among all the video compression schemes, digital pulse code module/discrete cosine transform (DPCM/DCT) compression is the most popular due to its capability to remove spatial and temporary redundancy to achieve low bit rate. Also it is more suitable for VLSI implementation. In DPCM/DCT compression scheme, a motion estimation module is required to estimate the motion vector of a particular macro-block. Instead of transmitting the compressed pixel data, only the motion vector and the prediction error are coded and transmitted. This greatly reduces the temporal/spatial redundancy and the number of bits required to represent the macro-block. Among all the signal processing modules required for video compression, motion estimation module is the most computational intensive (more than 50% of the entire system[5]), and hence consumes the largest amount of power.

For portable video applications, there are several design parameters that need to be trade-off: the bit rate, the algorithmic performance (or picture quality), the circuit complexity and the power consumption. If we use a very accurate predictor in the DPCM loop which gives very accurate motion estimation, the prediction error input to the DCT will be small. Thus we can either reduce the bit rate under fixed quantization step size or use smaller quantization step size under a constant bit rate requirement to achieve a better picture quality (In this paper, a constant bit rate requirement is assumed). However, accurate prediction needs more computation and hence higher complexity and power consumption. The relationship between the picture quality and the prediction accuracy is not a linear function. Hence it will be benefit to trade-off prediction accuracy for power consumption without a significant impact on the final picture quality.

There are different ways to reduce the complexity and the power consumption of the motion estimation. Approximate algorithms such as three-step search block matching algorithm can be used instead of full search block matching algorithm. Smaller search window can be used to reduce the computational load and power consumption [3]. Another method is using bit truncation technique in which the bit precision is adjusted such that the bit-width and thus the hardware cost of the motion estimation module are minimized while the performance still satisfies the requirement. In [6], a fixed bit-truncated block-matching motion estimation algorithm is proposed to reduce the hardware cost of the motion estimation module. It is shown that the incoming image pixel value can be truncated by 4 without significant degradation in the algorithm performance which is measured by the peak signal-to-noise ratio (PSNR) of the images. In this paper, we propose a bit masking technique to reduce power consumption of the motion estimation. Instead of using a fixed truncation, an adaptive bit masking method is used to determine the number of truncated bit in real time depending on the quality of the picture. If the accuracy is too low, the prediction error will accumulate and the magnitude of data feed to the quantizer is large. To maintain a constant bit rate requirement, a large quantization step size is required which degrade

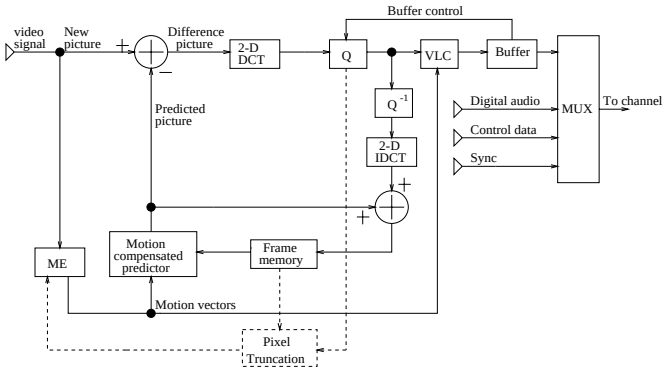

Figure 1: Overview of a DPCM/DCT based encoder

the picture quality. In the proposed adaptive scheme, the number of truncated bit will then be reduced by masking fewer LSB bits. By doing so, a better trade-off between power consumption and picture quality can be achieved.

The paper will be organized as follows. DPCM/DCT based video compression algorithm and block-based motion estimation algorithm will be reviewed in Section 2. Section 3 discusses the issues of fixed bit truncation and power consumption. The proposed adaptive bit masking method is presented in Section 4. Results on the power reduction and algorithmic performance are discussed in Section 5. Finally, we give the conclusion in Section 6.

II. DPCM/DCT based video compression algorithm and motion estimation algorithms

The most popular video encoders for standards such as MPEG1, MPEG2, H.261 or H.263 employ a DPCM transform-based compression technique which consists of the following basic functions: discrete cosine transform (DCT), quantization, variable length coding (VLC), bit-rate control, inverse discrete cosine transform (IDCT), dequantization, and motion compensation. Fig. 1 shows an overview of a DPCM/DCT based encoder. The block indicated by dash line is for the adaptive bit truncation which will be discussed in Section 4.

The main idea of the transform based video compression algorithm is to reduce the spatial redundancy within picture (intra-frame coding) and the temporal redundancy between pictures (inter-frame coding). Redundancy is exploited by a combination of motion-compensated differential pulse coded modulation (MC-DPCM) and transform coding. One important feature of the MC-DPCM loop is the motion compensated prediction. An important component in motion compensated prediction is the estimation of motion vector in successive frames.

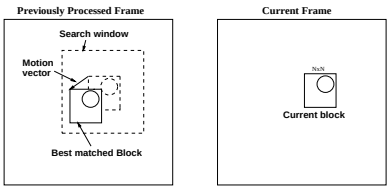

Figure 2: Motion estimation algorithm

Block-matching motion estimation technique is widely used to determine the motion vector which is the relative position between the current block and the best matched block in the search window in the previous processed frame (see Fig. 2). Mean absolute difference (MAD) is commonly used as the matching criterion and is calculated as follows:

$$MAD(m,n) = \sum_{i=1}^{N}\sum_{j=1}^{N} |x_{i,j} - y_{i+m,j+n}| \qquad (1)$$

where N is the block size, $x_{i,j}$ and $y_{i,j}$ are the pixel values of the current block and the previous block, respectively. The best matching is the block in the search area which has the minimum MAD. Since block matching algorithms (BMA) are computationally very demanding, application-specific VLSI implementations for BMA are required to satisfy the computational requirement. Many architectures have been proposed for fixed block size, full search and fast search BMA[2][8].

The prediction error depends on how accurate the motion estimation predicts. Larger prediction error will lead to higher bit rate or greater quantization step size and quantization error. However the relationship between the final picture quality and the prediction accuracy is not linear. To achieve a higher prediction accuracy, we need more intensive computation in the motion estimation and hence more power consumption. At the same time, it may not necessary result in significantly better picture quality. Therefore one way to reduce power consumption is to trade-off the prediction accuracy.

There are several ways to trade-off the prediction accuracy. The first is to use different algorithms. For example, fast search BMA such as three-step search, which is much simpler, can be used instead of full search BMA. The other way is to use a coarser pixel resolution to reduce the computation complexity. These two methods are orthogonal and can be combined together. In this paper, we will explore the power/picture-quality trade-off using coarser pixel resolution. In particular, we propose a variable bit resolution scheme in which the trade-off can be done adaptively with respect to the picture quality and the power consumption in real time. In the following sessions, we will discuss the technique of fixed bit truncation and adaptive bit masking, respectively.

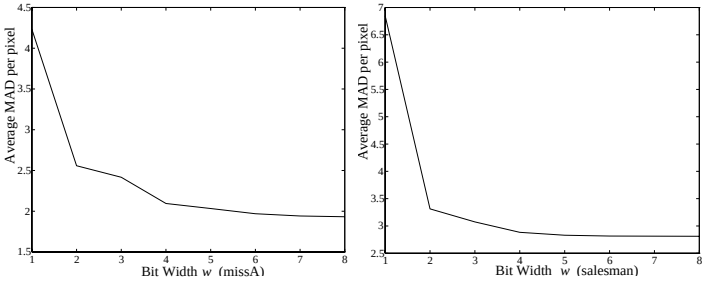

Figure 3: Average MAD per pixel vs. Bit Width

Sequences	$NTB = 4$ ($w = 4$)			$NTB = 3$ ($w = 5$)		
	max diff.	aver. diff.	MAD % inc.	max diff.	aver. diff.	MAD % inc.
missA	0.19	0.16	7.76%	0.11	0.10	4.97%
Sales.	0.09	0.07	2.49%	0.03	0.02	0.66%

Table 1: The error of MAD with $NTB = 4$ and $NTB = 3$

III. Fixed bit truncation

In this session, we investigate the power consumption reduction of the motion estimation module using fixed bit truncation on the pixel values. Full search algorithm is used to demonstrate the concept while it is also applied to other block matching ME algorithms. Simulation results show that the algorithmic performance of the block matching motion estimation, which is measured by the picture peak signal-to-noise ratio (PSNR), only degrades slightly when the LSBs of the pixel value are truncated.

It is difficult to decide the optimal number of truncated bit (NTB) analytically. However this can be determined from empirical data. We simulated several benchmark video sequences and looked at the impact of NTB on the calculation of MAD. We use the sequences of "missA" and "Salesman" for illustration. In our simulation, the first 150 frames of the video sequence were used. The simulation results of different NTB are shown in Figure 3. It is observed that the MAD decreases exponentially and it becomes flatten when the NTB is equal to 4. Table 1 summarizes the maximum MAD difference, average MAD difference and the % increase in average MAD with respect to the full precision (8-bit) FS-BMA for NTB equal to 3 and 4. Based on the above simulation results, NTB equal to 4 is a reasonable number to use.

Truncating the LSBs leads to less hardware and hence reduces the power consumption. Moreover, the switching activity is further reduced because of the nature of the video data. In [7], multimedia data is modeled by a Dual-Bit Type Model (DBT) of which the bits of the data are classified by two regions according to their switching

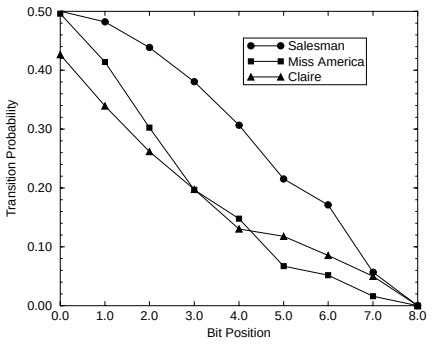

Figure 4: Bit transition for input of motion estimation.

activities: the low order bit region in which the bits are relatively independent and have higher switching activities, and the high order bit transition region in which the bits are highly correlated and have lower switching activities. Figure 4 shows the profile of the switching activity at the input of the motion estimation processing elements which closely resembles the DBT. The LSBs of the image data have much higher switching activities than the other bits. By truncating the LSBs of the pixel data, the percentage of switching activity reduction is higher than that of the bit truncation. Therefore further power reduction is achieved in addition to the reduction in complexity.

IV. Variable Truncated Bits using Adaptive Bit Masking

For some video sequences and applications where picture quality highly depends on the prediction error, a higher pixel resolution is preferred to achieve a lower prediction error. Hence the NTB should be reduced to achieve a better prediction. On the other hand, for some video sequences of which the sensitivity of the picture quality on prediction error is low, we can have a larger NTB to achieve a larger power reduction. Therefore it is better to have a scheme that the NTB can be changed in real time to cope with the picture quality requirement. Consequently, better trade-off between the power-consumption and algorithmic performance can be achieved.

In the following, we present a variable bit truncation algorithm using adaptive bit masking method. Instead of using PE elements with smaller bit width in the motion estimation datapath as in the fixed NTB scheme, PE elements that can accommodate full pixel resolution are used. The current and previous frame pixel data are sent to a mux before going into the motion estimator. The mux will select a zero to mask off the bits which are chosen to be truncated. By having a real-time adaptive control mechanism, the number of bit truncated can be

varied between different frames or different macro-blocks. To reduce the overhead of the computation complexity and power consumption, it is better to have a simple mechanism for the NTB control. We developed a simple model which adaptively truncate the number of bits based on the quantization step size.

Generally speaking, the quantization step size is determined by the error image of the original picture and the predicted picture with the consideration of buffer control. The complexity of the error image is related to the complexity of scenery and the accuracy of the prediction. Normally, sceneries with higher complexities or large motions will result in larger error image and consequently result in larger quantization step size, and vice versa. If we perform the truncation operation on the pixel values, the complexity of the error image will also be changed accordingly. As illustrated in Figure 3, the amplitude of error image is increased when the number of truncated bits increased. The bit rate control mechanism will change the quantization step to cope with the change in error image. In general the quantization step size is increased to maintain the constant bit rate requirement. Therefore, the quantization step size can be used to reflect the effect of the truncation operation.

The block diagram of the modified DPCM/DCT video encoder using the proposed adaptive bit masking approach is illustrated in Figure 1. The adaptive bit masking uses a feedback loop which is embedded in the DPCM loop. If we mask and truncate more bits, the error image is increased and the quantization step size will probably become larger. If the quantization step size becomes too large, we should reduce the number of masked/truncated bits to reduce the error image. Otherwise, we can maintain or further increase the number of masked/truncated bits while the quantization step size is unchanged or only slightly increased.

Based on the above description, we establish a simple model to adaptively change the NTB. It is known that the quantization step size can be changed on the frame-level as well as on block-level. Our proposed NTB control algorithm is based on the frame-level model. The algorithm is shown in Table 2. QP_{new} is the average quantization step size of current processed frame, QP_{mean} is the average quantization step size of the previous processed frames, and NTB is the number of truncated bits. The two parameters f_1 and f_2 is used to fine-tune the window of the quantization step size in which the NTB is remained constant.

When $QP_{new} \leq QP_{mean} \times f_1$ $(f_1 \geq 1)$, it means that the quantization step size is not particularly large and we can afford to have a larger prediction error. If NTB is less than an upper-bound (6 in our case), NTB is increased by 1. In this case, the complexity of the error image may be increased. This may cause an increase in the quantization step size of the next frame. NTB will keep on increasing as long as $QP_{new} \leq QP_{mean} \times f_1$ and NTB does not reach the upper bound. When the quantization step size is increased such that $QP_{new} > QP_{mean} \times f_2$ $(f_2 > f_1)$, it means the current quantization step size is too large and should be decreased. If the number of truncated bits is larger than a lower-bound (1 in our case), we should decrease the quantization step size by reducing the complexity of error image. This is implemented by reducing the NTB. When NTB has reached the low-bound or QP_{new} is reduced to the level close to QP_{mean} (i.e. $QP_{new} < QP_{mean} \times f_2$), NTB is remained unchanged.

If $(QP_{new} \leq QP_{mean} \times f_1$ and $NTB < 6)$
$\quad NTB = NTB + 1;$
Else if $QP_{new} > QP_{mean} \times f_2$ and $NTB > 1)$
$\quad NTB = NTB - 1;$
Else
$\quad NTB$ remains unchanged;

Table 2: The proposed model

V. Experimental results

We carried out simulations based on the H.263 standard which is implemented by Telenor Research [9]. Three video sequences were used for the experiment. They are the "missA", "Salesman" and "Claire" sequences. The "missA" and "Claire" sequences are slow moving while the "Salesman" sequence is of moderate moving. A constant bit rate of 32 kbps is assumed. The basic configuration of the H.263 is used, i.e., the four negotiable coding options (unrestricted motion vectors, syntax-based arithmetic coding, advanced prediction and PB-frames) are not included.

16×1 semi-systolic array architectures implementing the full search algorithm [2] are used for the comparison of power consumption. All the power consumption are estimated using an architectural power estimation framework [1] which is based on a cycle-based power macromodel [4]. Cycle-based power macromodel is built for individual processing element. Switching activities at every cycle are collected for every module during the architecture simulation. Power consumption is obtained by invoking the cycle-based power macromodel.

For fixed NTB, the variation of power among the frames is quite steady as seen in Figures 5. The change of power consumption is within 2% across 150 frames sim-

ulated. This is true for all the video sequences. Table 3 summarized the results on power consumption and PSNR for different fixed NTB values. It can be shown that the power consumption is reduced by an average 58% when NTB=4 since the most active 4 LSBs are truncated. At the same time the degradation in PSNR is less than 1%.

For variable NTB scheme, we used 1.0 and 1.09 for f_1 and f_2, respectively for the bit truncation control mechanism. These values are determined empirically. The initial number of truncated bits is set to 4 for the first inter-frame coding.

Figure 6 shows the simulation results on PSNR for "missA", "Salesman" and "Claire" test sequences, respectively. Three cases are compared in the simulations. They are full precision (FP), adaptive bit truncation (ABT) , and 4-bit fixed-bit truncation (4-bit FBT). From Figure 6(a), it can be shown that the PSNRs of the decoded pictures of the ABT scheme is better than that of 4-bits FTB scheme in most of the frame for the "missA" sequence. At the same time, the average NTB is 4.2 which is a little bit greater than 4. For the "Salesman" sequence, the PSNRs of the decoded pictures are very close for all the three cases. However, the average NTB of the ABT scheme is 5.7 which is much higher than the other two cases. That means on average only 2.3 bits per pixel are used for computation. For the "Claire" sequence, the PSNRs is a little bit worse than 4-bit FBT. However the average NTB is 5.75 which is a lot higher.

The observation of "missA" having a smaller average NTB than "Salesman" coincides with the results shown in Figure 3 for the average MAD simulation. When NTB equals 5 or 6, the % increase in average MAD over that with full precision for "Salesman" is small, while for "missA", there is a sudden increase in the % increase of MAD when NTB is increased from 4 to 5. Therefore, for "Salesman", even $NTB = 5.7$, the performance is close to that of full-precision (see Fig. 6(b). Similar explanation can be given for "Claire" sequence.

Figures 5 also shows the variation of power consumption among different frames which depends on the number of bits truncated in each frame. It shows that the NTB range from 2 to 6 for "missA" and is most of time lower than 5 for the other two sequences. Table 4 summarizes the comparison of PSNR and power consumption for FP, ABT and 4-bit FBT. It can be shown that the power consumption of ABT for the "Salesman" and "Claire" sequence is reduced by more than 70% and 20% compared with the full precision and the 4-bit FBT respectively. It is because in a lot of frames the NTB can be increased to 6 and the average NTB are equal to 5.7 and 5.75, respectively. For the "missA" sequence, the power consumption of ATB is a little bit higher than that of 4-bit FBT while the average PSNR is higher. It is because in a lot of frames, the NTB has to be set to 2 in order to lower the prediction error so to reduce the quantization step size. In general, ABT gives a better and more flexible power/performance trade-off than a FBT scheme.

VI. Conclusion

In this paper, we investigated the trade-off between the prediction accuracy and power consumption in the motion estimation module. We proposed a fixed bit truncation scheme and an adaptive bit masking technique to reduce the power consumption of the motion estimation. while maintaining the picture quality. Experimental results showed that over 70% reduction in power consumption can be achieved without significantly degrading the PSNR.

Acknowledgement: This work was supported in part by Hong Kong RGC Research Grant HKUST779/96E, and by the Hong Kong Telecom Institute of Information Technology Grant HKTIIT92/93.001.

References

[1] K.K. Chan and C.Y. Tsui. Exploring the power consumption of different motion estimation architectures for video compression. *Proceedings of IEEE ISCAS'97*, June 1997.

[2] K. M. Yang et al. A family of vlsi designs for the motion compensation block-matching algorithm. *IEEE Trans. on Circuits and Systems*, 36(10):1317–1325, Oct. 1989.

[3] Masayuki Mizuno et. al. A 1.5w single-chip MPEG MP@ML encoder with low-power motion estimation and clocking. *Proceedings of IEEE ISSCC*, pages 256–257, Feb. 1997.

[4] Q. Wu et al. Statistical design of macro-models for rt-level power estimation. *Proceedings of ASP-DAC*, pages 523–528, Feb. 1997.

[5] K. Guttag, R.J. Gove, , and J.R. Van Aken. A single chip multiprocessor for multimedia: The mvp. *IEEE Computer Graphics and Applications*, pages 53–64, Nov. 1992.

[6] Zhong L. He and Ming L. Liou. Reducing hardware complexity of motion estimation algorithms using truncated pixels. *Proceedings of IEEE ISCAS'97*, June 1997.

[7] P. Landman J. Rabaey. Power estimation for high level synthesis. *Proceedings The European Conference on Design Automation*, pages 361–6, 1993.

[8] N. Demassieux P. Pirsch and W. Gehrke. Vlsi architectures for video compression—a survey. *Proceedings of IEEE*, (2):220–246, Feb. 1995.

[9] Telenor R&D. H.263 simulator based on the tmn5. Technical report, 1996.

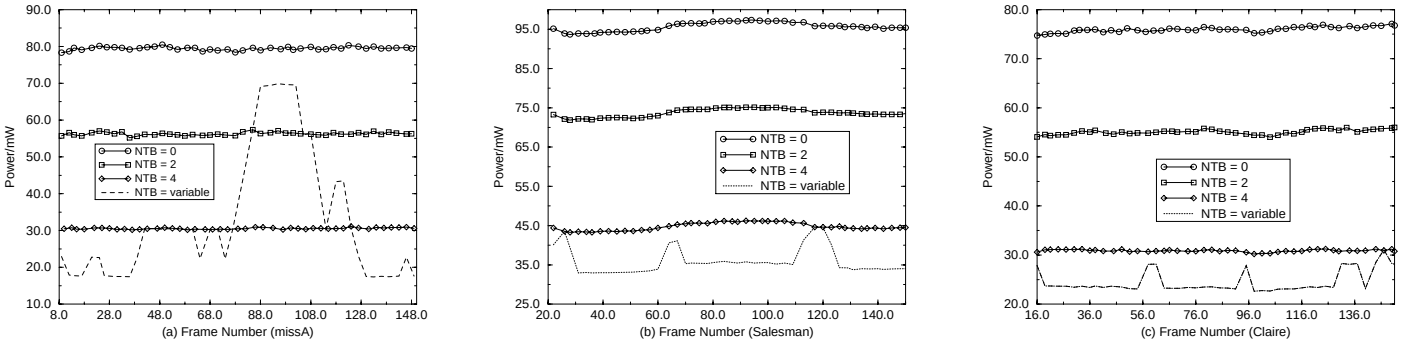

Figure 5: Power Consumption of (a) "missA", (b) "Salesman", and (c) "Claire" sequences with different NTB

	Power/mW			PSNR/dB		
NTB	"Salesman"	"missA"	"Claire"	"Salesman"	"missA"	Claire
0	95.68	79.46	75.97	32.677	40.117	39.376
2	73.69	56.27	55.08	32.670	40.071	39.301
4	44.75	30.55	30.88	32.678	39.972	39.345

Table 3: The power consumption and PSNR for Fixed NTB

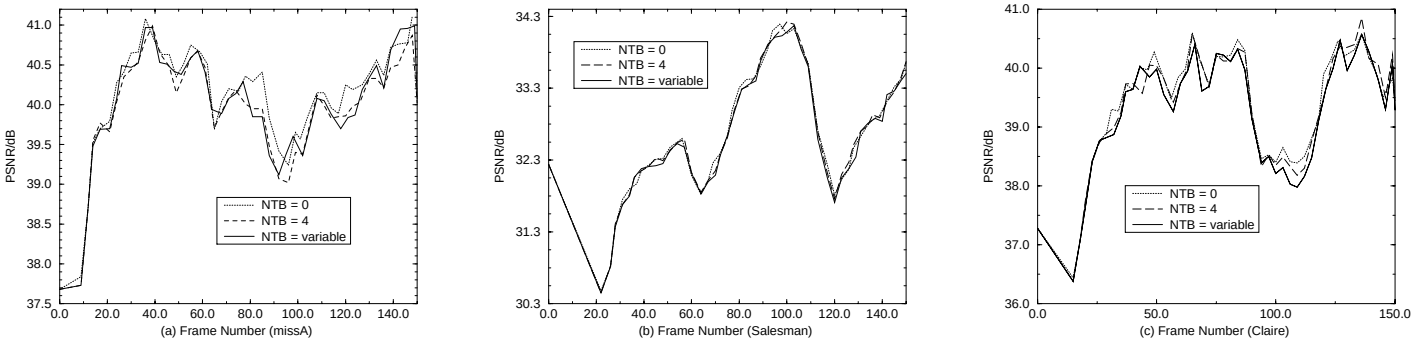

Figure 6: PSNR simulation for (a) "missA", (b) "Salesman", and (c) "Claire"

	Power/mW			Aver. PSNR/dB		
NTB	"Salesman"	"missA"	"Claire"	"Salesman"	"missA"	Claire
0	95.68	79.46	75.97	32.677	40.117	39.376
4	44.75	30.55	30.88	32.678	39.972	39.345
adaptive	35.84	32.59	24.55	32.645	40.019	39.246

Table 4: The result of power consumption and PSNR for ATB