Holter ECG Compression using DSP

Shah Anup Ushakant and A. G. Ramakrishnan

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012

Abstract - TMS320C31 starter-kit is used to implement ECG compression in real-time. The ECG data-stream is segmented on R-waves. After interpolating the delineated beats using discrete cosine transform, period and amplitude normalized beats are vector quantized. The algorithm is optimized for real-time implementation. We obtained a compression ratio of 120 with normalized root mean square error of about 2%.

I Introduction

The electrocardiogram (ECG), which represents the electrical activity of the heart, is vital for the diagnosis of most cardiac disorders. To record one patient's ECG for a duration of 24-hours, the storage requirement ranges from 34 Mbytes for a single lead ECG sampled at 250 Hz and 12 bit resolution, to 175 Mbytes for two-leads ECG sampled at 500 Hz and 16 bit resolution. For efficiently managing storage space, ECG compression is a must. In Ambulatory ECG monitoring, ECG compression needs to be done in real time.

In this work, an ECG compression scheme is implemented in real time on the TMS320C31 floating point Digital Signal Processor. The input continuous-time signal is sampled at 12-bit resolution at 250 Hz sampling rate.

II Algorithm

If we can make the beat period uniform for all beats, the performance of beat-based compression schemes will improve.

Period Normalization : The idea of period normalization (PN) is to resample the signal with different sampling rates for different beats, such that the length of each beat becomes a pre-fixed number, say, L_N. The following procedure achieves this end.

- Take L_i length DCT of x[n]
- Pad (L_N - L_i) zeros to DCT of x[n] to make the length L_N.
- Multiply by $\sqrt{L_N/L_i}$ to compensate for different lengths in DCT and IDCT.
- Take L_N length IDCT and get the PN sequence $x_N[n]$ of length L_N.

Amplitude Normalization : The amplitude of the peak value (R point) varies from beat to beat. After period normalization, we can improve the inter-cycle correlation further by performing amplitude normalization. The idea of amplitude normalization is to scale the PN-beat by a factor such that, the magnitude of the sample having the highest value in the beat becomes a fixed value. As ECG is sampled at 12-bit resolution, we choose to scale the highest value to 4095 and accordingly scale all other samples.

In period normalization process, we have to scale the vector to compensate for the difference between the lengths of DCT and IDCT. Instead of scaling twice (during period normalization and amplitude normalization), we can perform scaling once finally, as IDCT is a linear transformation.

For transmission, we code the difference between the average scaling factor and the scaling factor for the present PN beat. On reconstruction, the actual scaling factor can be derived using mean scaling factor, which is made available with the codebook.

III Implementation

Segmentation of ECG Beats : It is very important to accurately detect the R points and segment the ECG beats correctly. Incorrect detection of R point results in faulty reconstruction, which, in turn results in misinterpretation of reconstructed ECG.

The detection process is divided into two sections [1]. The preprocessor section performs linear and nonlinear filtering of the signal and produces a data-stream, which describes significant events. The decision rule section operates on the output of the preprocessor and classifies each event as either R point or not.

The process of R point detection is divided in to two phases: learning phase and detection phase. Learning phase requires 2 to 3 second data to initialize the detection thresholds. The subsequent detection phase includes a recognition process, which estimates the threshold and identifies the R point events. Threshold level is adapted on-line to make the recognition process more robust against changing characteristics of the signal. Search back is done in case R wave is not detected for sufficient long time.

Codebook Generation : Each of the period and amplitude normalized beats, obtained as explained above, is treated as a vector of dimension L_N. Codebook is designed using the modified Linde-Buzo-Gray algorithm.

Modified LBG Algorithm : In this approach, the steps involved in arriving at the codebook are as follows:

1. Initial part of the data is chosen as the training vector set after PAN.
2. The centroid of all the training vectors is initialized as a size 1 codebook.
$$C_1 = \frac{1}{N_{tr}} \sum_{i=1}^{i=N_{tr}} V_i \quad (5)$$
where, N_{tr} is total number of training vectors.
3. A new pair of codebook vectors corresponding to each codebook vector in the existing set is derived by adding and subtracting a fixed perturbation.
$$C_{2j} = C_j + P$$
$$C_{2j+1} = C_j - P \quad (6)$$
where P is a small non-zero perturbation.
4. The minimum distortion for the i^{th} training vector is calculated as,
$$D(i) = \min_j \| V_i - C_j \|_2 \quad (7)$$
here, C_j is the j^{th} codebook vector, j = 1 to S, where S is the present size of codebook. The distortion is calculated for each of the training vectors.
5. Each codebook vector is updated by the centroid of all the training vectors mapping to it.
6. The total distortion for the current iteration is obtained as,
$$D_{total} = \sum_{i=1}^{i=N_{tr}} D(i) \quad (8)$$
7. The value of total distortion D_{total} is compared with that of the previous iteration. If the reduction in distortion is less than a certain threshold and the present codebook size is less than the desired codebook size, jump to step 3. Otherwise jump to step 4 if the reduction in the total distortion is higher than the fixed threshold.

This algorithm generates the codebook of the required size.

Encoding and Decoding : The codebook obtained is now made available at both the encoder and the decoder. When a new ECG beat is received, it is period and amplitude normalized. Distortion of this PAN beat with respect to each of the codebook vectors is calculated. The index of the codebook entry having minimum distortion is stored as the code for the present ECG beat. Along with the index, the differential period and amplitude-normalizing factors are also stored. Thus, for each normalized beat vector, there are only $\log_2 S$ (S being the size of codebook) plus bits for encoding differential period and differential amplitude scaling factor are required. At the decoder end, the actual normalizing factors are derived from the mean amplitude and period normalizing factors, made available along with codebook, and the present differential codes. Using these factors, the ECG beat with the original amplitude and period is reconstructed.

Hardware Design : The period & amplitude normalization, together with vector quantization of PAN beat requires around 0.4 MFLOPS. In addition to that, for reliable R point detection, digital filters, threshold estimator and the detection logic routines, which require far more integer operations than floating point operations, must run at the sampling rate. So we have chosen the floating point DSP TMS320C31 ('C31) [3] for implementing the compression scheme in real time.

Starter Kit : The starter kit [1] of 'C31 provided by Texas Instruments has on-board Analog Interface Circuit (AIC) TLC32040CN, interfaced to computer through parallel printer port and 2K on-chip RAM [3]. The AIC provides:

- a single channel input/output analog interface with 14-bit ADC and DAC
- a sampling rate of upto 20 kHz.

The AIC is connected to 'C31 through serial port 0 of DSP. For resetting AIC, XF0 pin is used, while timer 0 of 'C31 drives clock of AIC.

The personal computer communicates with the 'C31 through the parallel printer port. The printer port is mapped to the address range 0xFFF000-0xFFFFFF of 'C31. In one of the possible four boot-loader mode, (see [3], section 3.4), DSP can be booted by downloading code from the computer.

Expansion Board : For implementing the compression scheme, program-memory and additional data-memory are interfaced to the starter kit through four *expansion connectors* provided on the kit [1]. The 32 K EPROM is mapped to the address space 000000-007FFFh address. This falls in Boot 1 region of 'C31. The 32 K static RAM is mapped to the address space 700000-707FFFh.

The option of boot-loading from address range 0xFFF000-0xFFFFFF of 'C31 must be disabled to boot-load from EPROM on expansion board. This can be achieved by tri-stating programmable array logic (PAL) TICPAL22V10Z. The DSP reset signal, tri-state PAL signal and enable boot-load from EPROM signal are jumper selectable on expansion board.

Software Optimization : In order to reduce development-cycle in possible reusability of code we have used C routines. But the cross compiler generated code was not optimal to do compression in real-time. So we have located those frequently called routines where compiled code was far from optimal. We rewrote those routines in assembly optimally. The compiler was not able to use features like circular addressing mode and parallel-instructions of DSP many times.

The hardware configuration was fit for big memory model of compiler. But we have compiled different modules of code using small memory model option and added code for loading Data Page where-ever required.

For period normalization we need to implement DCT routine in the most general way, as beat length keeps changing to odd values. We need to compute different values of cosine function for each beat. We have used caching for cosine function values in order to avoid possible reuse of previously calculated function values.

IV Performance Measures

We have evaluated the performance of the technique by computing the compression ratio (CR), the Normalized Root Mean Square Error (NRMSE) and Normalized Maximum Amplitude Error (NMAE).

V Conclusion

For implementing the real-time compression scheme, we have designed a memory expansion board, having 32k single wait-state static RAM and 32k EPROM. An user selectable option is given on the expansion board to boot-load the DSP either from the PC or from the on-board EPROM. The communication kernel of the 'C31 starter-kit is modified to simulate the real-time ECG data input through parallel printer port. However, the communication protocol between the starter-kit and the personal computer is not affected.

The previously recorded ECG data-stream was fed at 250 Hz. The incoming data-stream was delineated on every R-wave. The variable length beats are period and amplitude normalized. Subsequently, the PAN beats are vector quantized. The index of codebook vector, along with the differential period and the differential amplitude-scaling factor were stored in RAM on the expansion board. Subsequently, the compressed data was up-loaded to the computer and reconstructed for checking on the fidelity of reconstruction. While reconstructing on the personal computer, 32-bit single precision arithmetic was used to obtain results same as what can be achieved on DSP. The compressed data can also be reconstructed in real-time by the DSP.

A CR of upto 120 was achieved with a codebook of size 16 with very low reconstruction error. The R-wave detection, PAN and VQ are implemented on DSP 'C31 with real-time constraints.

Table 1

Performance figures (16-size codebook)

Sr. No.	CR	NRMSE %	NMAE %
1	93	3.53	13.61
2	119	1.45	11.4
3	97	1.78	17.4
4	103	1.45	9.01
5	80	1.44	10.14
6	102	0.88	7.66

VI References

1. Patrick S. Hamilton & Willis J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database", IEEE Transaction on Biomedical Eng., Vol. BME-33, No. 12 pp:1157-1165, December 1986.
2. Texas Instruments Digital Signal Processing Products, "TMS320C3x User's guide", 1994.
3. Texas Instruments Digital Signal Processing Solutions, "TMS320C3x DSP starter kit" - User's guide", 1996.