프랙탈 알고리즘 기반의 실시간 영상 부호화기의 설계 및 구현

김 재 철† · 최 인 규††

요 약

이 논문에서는 범용 DSP칩인 ADSP2181를 사용하여 프랙탈 알고리즘 기반의 영상 부호화기를 설계 제작하였다. 제작된 부호화기는 고정소수점을 지원하는 Analog Device사의 ADSP2181 두 개를 사용하여 구현되었고, 영상부호화는 3단계의 파이프라인 구조에 의해 이루어진다. 첫 번째 파이프라인단인 영상 획득부는 NTSC 표준 영상 신호로부터 디지털 영상 데이터를 획득하여 프레임 메모리에 저장한다. 두 번째단에서의 주제어부에서는 영상 데이터를 프랙탈 알고리즘을 이용하여 부호화를 수행한다. 마지막 단인 출력제어부는 부호화된 영상 계수를 RS422 포트를 통하여 출력하도록한다. 설계 제작된 프랙탈 영상 부호화기의 성능은 QCIF 영상 포맷에서 정지영상에 대하여 초당 10프레임 이상의 부호화 속도를 얻었다. 프랙탈 알고리즘을 이용하여 프레임간 중복성을 이용한 영상 부호화시에는 초당 평균 30 프레임 이상의 부호화속도를 얻을 수 있었다.

Design and Implementation of Real-time Moving Picture Encoder Based on the Fractal Algorithm

Jae Chul Kim† · In Kyu Choi††

ABSTRACT

In this paper, we construct real-time moving picture encoder based on fractal theory by using general purpose digital signal processors. The constructed encoder is implemented using two fixed-point general DSPs (ADSP2181) and performs image encoding by three stage pipeline structure. In the first pipeline stage, the image grabber acquires image data from NTSC standard image signals and stores digital image into frame memory. In the second stage, the main controller encode image dada using fractal algorithm. The last stage, output controller perform Huffman coding and result the coded data via RS422 port. The performance tests of the constructed encoder shows over 10 frames/sec encoding speed for QCIF data when all the frames are encoded. When we encode the images using the interframe and redundency based on the proposed algorithms, encoding speed increased over 30 frames/sec in average.

키워드 : 프랙탈(fractal), 부호화기(encoder), 영상 부호화(image coding), 영상획득부(image grabber), 주제어부(main controller), 출력제어부(output controller), 디지털신호처리칩(DSPs)

1. 서 론

멀티미디어 시대가 도래함에 따라 영상 정보의 저장과 전송에 대한 요구가 증가하고 있으나 영상 데이터는 그 양이 방대하여 아무런 처리없이 이를 저장, 전송하는 것은 생각할 수가 없다. 특히, 대역폭이 제한된 채널을 통하여 영상 정보를 전송할 경우에는 데이터의 압축은 필수적이다. 여러 압축기법가운데 프랙탈 이론을 이용한 영상 부호화 기법은 영상 내에 존재하는 자기유사성을 이용하여 중복성을 제거하는 방법[1]으로 Jacquin[2, 3]에 의해 처음 제안된 이래 Oien[4], Monro[5] 외 여러 연구자들에 의해 많은 연구결과[6-9]가 보고되어 왔다. 프랙탈 영상 압축에서는 영상을 고정 크기 혹은 가변 크기의 블록으로 나눈후 부호화하고자 하는 블록이 반복적인 축소 변환에 의해 초기 영상에 무관하게 고정점으로 수렴하는 성질을 이용하여 이때의 축소 변환계수로 부호화한다. 이러한 부호화 방법에서는 영상 내에서 부호화하고자 하는 블록과 가장 유사한 블록을 찾아 축소 변환 계수를 구해야하므로 탐색으로 인한 계산량이 방대하여 실시간 처리기의 구현이 어렵고 하드웨어에 대한 연구가 미진한 상태이다.

본 논문에서는 실시간 부호화를 이루기 위하여 요구되는 계산량, 데이타 압축비 및 신호대 잡음비의 적합화를 위하여 기존에 발표된 정지영상의 부호화시 적용되었던 알고리즘을 수정하여 결합한 확장된 타일링 구조와 수정된 쿼드-트리 분할 알고리즘[10]을 이용한 실시간 프랙탈 전용 부호화 시스템을 설계, 제작하였다.

본 논문에서 제작된 실시간 프랙탈 부호화기는 범용 DSP

※ 이 논문은 2001년도 구미1대학의 연구비에 의하여 연구되었음.
† 정 회 원 : 구미1대학 정보통신전공 교수
†† 정 회 원 : 경북대학교 대학원 전자공학과
논문접수 : 2002년 6월 7일, 심사완료 : 2002년 10월 8일

를 이용하여 설계 제작되었고, 이는 QCIF 형태(144×176)의 동영상에 대해 평균 초당 30프레임 이상의 부호화 속도를 갖는다. 설계 제작된 부호화기는 고정 소수점 DSP인 ADSP-2181[11] 2개를 포함하여 3단계로 이루어지는 파이프라인 동작을 하도록 설계하였고 설계된 디지탈 회로에 대한 검증은 MicroSim[12] 프로그램을 이용하여 시뮬레이션 하였다.

본 논문의 2장에서는 프랙탈 기반의 실시간 처리 영상부호화 알고리즘에 대하여 서술하고 3장에서는 제작된 부호화기의 하드웨어 구조에 대해 설명한다. 그리고 4장에서 제작된 실시간 프랙탈 동영상 하드웨어의 성능을 분석하고 5장에서 결론을 맺는다.

2. 프랙탈 이론

2.1 수학적 배경

프랙탈 이론[13, 14]은 수학의 한 분야인 위상기하학(topology)에서 유래된 이론으로 척도(metric) 공간, 축소(contraction) 변환, 유사(affine) 변환 등을 그 기초로 하고 있다.

척도 공간이란 다음의 조건

$$d(x,y) = d(y,x) \tag{2-1}$$

$$0 < d(x,y) < \infty, \quad x \neq y \tag{2-2}$$

$$d(x,x) = 0 \tag{2-3}$$

$$d(x,y) \leq d(x,z) + d(z,y) \tag{2-4}$$

들을 만족하는 척도 함수 d를 갖는 공간을 의미하며 유사 변환, 축소 변환 등은 모두 척도 공간상에서의 변환이다.

유사 변환은 확대, 대칭, 회전, 축소등의 형태 변환으로 척도 공간에서의 선형 변환과 이동(translation or shift)으로 보여지는 유클리드 2차원 공간(R^2) 상의 유사 변환을 나타내며, 영상 신호는 좌표 공간외에 화소값 g을 갖는 3차원 신호이므로 3차원으로의 확장이 요구된다. 이를

$$\begin{aligned} W^{(k)}(x,y,g) &= S^{(k)}(x,y,g) + T^{(k)} \\ &= \begin{bmatrix} M_x^{(k)}\cos\theta_x^{(k)} & -M_y^{(k)}\sin\theta_x^{(k)} & 0 \\ M_x^{(k)}\sin\theta_x^{(k)} & M_y^{(k)}\cos\theta_y^{(k)} & 0 \\ a_1^{(k)} & a_2^{(k)} & a_3^{(k)} \end{bmatrix} \begin{bmatrix} x \\ y \\ g \end{bmatrix} + \begin{bmatrix} T_x^{(k)} \\ T_y^{(k)} \\ b^{(k)} \end{bmatrix} \end{aligned} \tag{2-5}$$

와 같이 표현될 수 있다. 식 (2-5)에서 $S^{(k)}$는 축소 선형 변환이고 $T^{(k)}$는 위치 이동 변환이다.

Monro는 식 (2-5)을 최적화시켜 프랙탈 부호화를 수행하고 성능의 향상을 위하여 좌표 x, y에 대한 근사화 식의 차수를 높임으로써 4원 1차 연립 방정식의 해를 구하여 프랙탈 부호화를 수행하였다.

2.2 반복함수계 이론

완전 척도 공간(complete metric space) (M, d)에서 정의되는 변환 $W : M \rightarrow MX$에 대해

$$d(W(X), W(Y)) \leq s \cdot d(X,Y) \quad \forall X, Y \in M \tag{2-6}$$

을 만족하는 상수 $0 \leq s < 1$가 존재하면, 변환 W를 축소변환[15]이라 한다. 여기서 $d(X, Y)$는 두 영상 X, Y 사이의 거리를 나타내는 척도이고, s는 변환 W에 대한 축소비(contractivity factor)를 의미한다. 이러한 유한개의 축소 변환으로 이루어진 계를 반복함수계(Iteration Function System)라 한다.

축소변환 W는

$$W(X) = LX + T \tag{2-7}$$

로 나타내고 축소선형변환영상 LX와 영상신호 X에 무관한 평행변환 영상 벡터 T로 나눌 수 있다.

프랙탈 변환 W를 초기영상 X_0에 대하여 무한히 반복하여 얻어지는 끌개영상(attractor image) X_T는

$$X_T = (1 + L + L^2 + \cdots + L^{\infty})T + L^{\infty}X_0 \tag{2-8}$$

와 같다.

식 (2-8)에서 L의 놈(norm) $\|L\|$이 1 보다 작으면 우측항 $L^{\infty}X_0$은 없어지고 끌개영상 X_T는

$$X_T = (I - L)^{-1}T \tag{2-9}$$

와 같이 표시된다.

식 (2-9)에서 I는 단위행렬(identity) 연산자를 의미한다. 따라서 복호화시 임의의 초기영상을 이용하여 프랙탈 변환을 무한히 반복하면 초기영상과 무관한 끌개 영상을 얻게된다.

콜라쥬(collage) 이론[3]에 의하면 끌개영상 X_T와 원 영상 X_{orig}와의 거리는 다음의 관계

$$d(X_{orig}, X_T) \leq \frac{1}{1-s}\, d(X_{orig}, W(X_0)) \tag{2-10}$$

를 만족시킨다.

2.3 타일링(Tiling) 구조와 프랙탈 변환

(그림 2-1)은 타일링 구조[12]와 프랙탈 변환이 수행되는 과정을 표시한다. 이 구조는 원 영상을 $2B \times 2B$의 크기를 갖는 정의역 블록 D로 나누고, 이를 가로 세로 각각 2등분하여 (그림 2-1)을

$$\begin{aligned} D &= \sum_{n=0}^{2B-1}\sum_{m=0}^{2B-1} d(n,m) = \bigcup_{k=1}^{4} R^{(k)} \\ &= \bigcup_{k=1}^{4}\sum_{p=0}^{B-1}\sum_{q=0}^{B-1} r^{(k)}(p,q) \end{aligned} \tag{2-11}$$

과 같이 $R^{(1)}, R^{(2)}, R^{(3)}, R^{(4)}$의 네 개의 치역블록으로 나눈다.

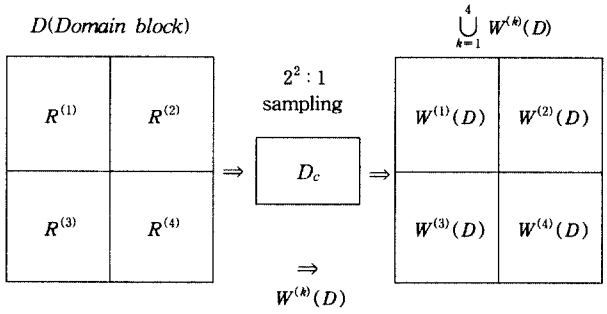
정의역 블록 D에서 각 치역 $R^{(k)}$(k=1,2,3,4)로의 프랙탈 축소 변환 $W^{(k)}(D)$는

$$\hat{R}^{(k)} = W^{(k)}(D) = s_k D_c + T^{(k)} \quad (2\text{-}12)$$

와 같이 표시된다. 여기서 식 (2-11)에서 $R^{(k)}$는 정의역 블록을 이루는 치역블록, 식 (2-12)에서 $\hat{R}^{(k)}$는 복원된 치역 블록 그리고 D_c는 $2^2:1$공간축소변환된 정의역 블록, s_k는 축소변환 상수를 표시하고 $T^{(k)}$는 평행 변환 연산자이다. 하나의 정의역 블록에서 영상의 부호화는 원 영상에 대하여 식 (2-12)에서 $R^{(k)}$에 최적으로 근사시키는 s_k값과 $T^{(k)}$의 파라메터 값을 구함으로 프랙탈 부호화가 이루어진다. 프랙탈 복원화는 콜라쥬 이론에 의해

$$W(D) = \bigcup_{k=1}^{4} W^{(k)}(D) = \bigcup_{k=1}^{4} \hat{R}^{(k)} \quad (2\text{-}13)$$

와 같이 표현된다.

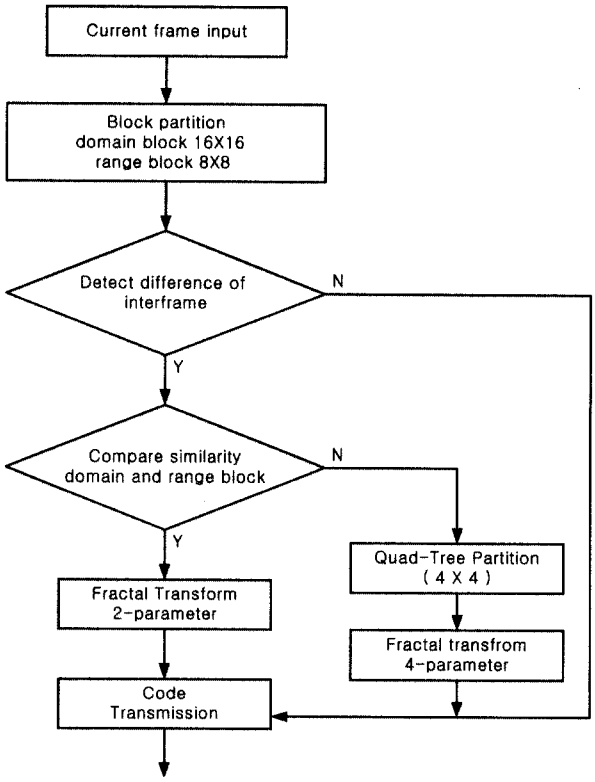


(그림 2-1) 타일링 구조와 프랙탈 변환

3. 동영상 부호화 알고리즘

본 논문에서는 부호화의 소요 시간에 중점을 두어 실시간 처리 가능한 프랙탈 동영상 부호화 알고리즘[10]을 이용하였다. 본 논문에서 이용한 부호화 과정은 움직임이 발견되는 부분과 움직임이 없는 부분으로 나누어 이루어진다. 움직임이 없는 부분은 이전의 영상을 이용하며, 움직임이 발견되는 부분은 프레임내 부호화를 한다. 프레임내 부호화에서 가장 큰 계산상의 부담인 탐색영역에 대한 시간을 최소화하기 위하여 탐색과정을 고정시킨 타일링(tiling) 구조[5]를 이용하였다. 부호화 계수는 최소자승오차(least square error)법을 이용하여 얻어진 다원 1차 연립 방정식의 해를 구하는 방법을 사용하였다[16]. 또한 비트율을 줄이기 위한 쿼드-트리 분할 방식은 정의역 블록과 치역 블록의 평균자승오차에 대한 평탄 임계치의 대소에 따라 프랙탈 계수의 개수를 변화시켰다.

(그림 3-1)은 제작된 부호화기에서 이용한 실시간 처리를 위한 프랙탈 부호화 알고리즘의 흐름도[10]이다. (그림 3-1)의 흐름도에서 연속적으로 입력되는 현재 프레임 f_n과 이전 프레임 f_{n-1}을 일정한 크기의 8×8 치역 블록들로 나누



(그림 3-1) 연속 영상을 위한 프랙탈 부호화 알고리즘의 흐름도

고 이들을 같은 위치의 블록에 대한 평균자승오차와 움직임 임계치 TH_m를 비교하여 오차가 작은 영역, 즉 움직임이 없는 영역은 부호화하지 않고 이전의 복원영상을 이용함으로써 비트율과 부호화 시간을 줄일 수 있었다.

블록의 분할을 위하여 이전 프레임 f_{n-1}에 대해 현 프레임 f_n의 움직임이 발견된 블록에서는 프레임내 부호화와 쿼드-트리 방식을 이용한다.

치역 블록 8×8 R_k에 대한 근사화은

$$d(R_k, W_k(R_k)) = \sum_{i=1}^{M}\sum_{j=1}^{N}\{R_k(i,j) - [b^{(k)} + a_3^{(k)} \cdot D_c(i,j)]\}^2 \quad (3\text{-}1)$$

와 같으며, 이는 공간축소 변환된 정의역 블록 D_c과 치역 블록 R_k의 평균 자승오차가 분할임계치 TH_s보다 작을 경우 적용된다.

평균 자승 오차가 분할 임계치 보다 클 경우는 4×4 블록으로 쿼드-트리 분할을 되며, 이 때 사용되는 부호화 근사화식은

$$d(R_{km}, W_{km}(R_{km})) = \sum_{i=1}^{M}\sum_{j=1}^{N}\{R_{km}(i,j) - [b^{(km)} + a_1^{(km)} \cdot x_i + a_2^{(km)} \cdot y_j + a_3^{(km)} \cdot D_{rc}(i,j)]\}^2 \quad (3\text{-}2)$$

이 적용된다.

치역 블록 8×8 R_k는 4×4 치역 블록의 정의역 블록 D_r로 되어 4개의 치역 블록 $\bigcup_{m=1}^{4} R_{km}$으로 된다. 쿼드-트리된 블록의 공간 축소변환된 정의역블록 D_{rc}는

$$D_{rc}(i,j) = R_k(2i, 2j) = D_r(2i, 2j) \tag{3-3}$$

와 같다.

치역 블록 8×8을 부호화하기 위하여 식 (3-1)을 최적화시킨 다원1차 연립방정식은

$$\begin{bmatrix} N & \sum_{i=1}^{M}\sum_{j=1}^{N} D_c(i,j) \\ \sum_{i=1}^{M}\sum_{j=1}^{N} D_c(i,j) & \sum_{i=1}^{M}\sum_{j=1}^{N} D_c(i,j)^2 \end{bmatrix} \begin{bmatrix} b^{(k)} \\ a_3^{(k)} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{M}\sum_{j=1}^{N} R_k(i,j) \\ \sum_{i=1}^{M}\sum_{j=1}^{N} R_k(i,j) D_c(i,j) \end{bmatrix} \tag{3-4}$$

와 같다. 그리고 쿼드-트리 분할된 4×4 치역블록을 부호화하기 위하여 식 (3-2)를 최적화시킨 다원1차 연립방정식은

$$\begin{bmatrix} N & 0 & 0 & \sum\sum D_{rc}(i,j) \\ 0 & N\sum x_i^2 & 0 & \sum\sum x_i D_{rc}(i,j) \\ 0 & 0 & M\sum y_j^2 & \sum\sum y_j D_{rc}(i,j) \\ \sum\sum D_{rc}(i,j) & \sum\sum x_i D_{rc}(i,j) & \sum\sum y_j D_{rc}(i,j) & \sum\sum D_{rc}(i,j)^2 \end{bmatrix} \begin{bmatrix} b^{(km)} \\ a_1^{(km)} \\ a_2^{(km)} \\ a_3^{(km)} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{M}\sum_{j=1}^{N} R_{km}(i,j) \\ \sum_{i=1}^{M}\sum_{j=1}^{N} x_i R_{km}(i,j) \\ \sum_{i=1}^{M}\sum_{j=1}^{N} y_j R_{km}(i,j) \\ \sum_{i=1}^{M}\sum_{j=1}^{N} R_{km}(i,j) D_{rc}(i,j) \end{bmatrix} \tag{3-5}$$

와 같다.

식 (3-4)와 식(3-5)의 좌변 행렬식의 값이 0을 가지면, 즉 다원 1차 연립 방정식이 선형 독립이 아니다. 이때 프랙탈 계수를 구하는 행렬의 연산은 불능이 되고 $a_3^{(k)}$의 절대값이 1에 근접하게 되면 복호화시 수렴속도가 현저히 느려지며 $a_3^{(k)}$의 절대값이 1보다 크면 발산이 된다. 그러므로 제작된 부호화기에서는 $a_3^{(k)}$의 절대값이 0.9 보다 클 경우는 0.5로 고정시키는 방법을 사용하였다[10].

4. 동영상 부호화기의 설계

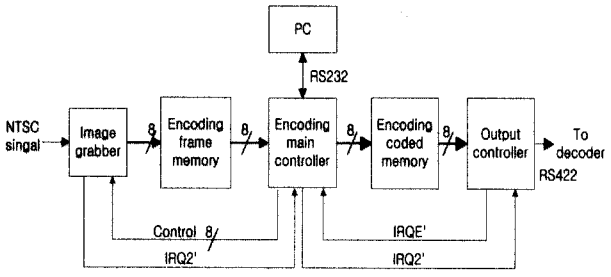
제작된 동영상 부호화기는 실시간 처리를 위한 처리 속도를 증가시키기 위하여 복수개의 프로세서를 사용하는 병렬 프로세서 구조로 이루어진다. 설계된 시스템의 수행능력을 향상시키기 위하여 파이프라인 동작을 하도록 하였다. 또한 RISC 구조의 프로세서에서 속도를 저하시키는 요인인 프로그램 길이의 증가를 피하기 위하여 모든 프로그램은 C 크로스-컴파일러(cross-compiler)[17]를 사용하지 않고 어셈블리어로 작성하였다.

4.1 동영상 부호화기의 하드웨어 설계

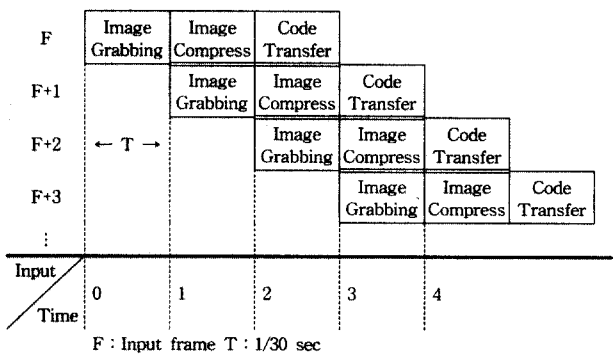
설계된 동영상 부호화기는 크게 영상획득부(image grabber), 주제어부(main controller), 출력제어부(output controller)로 나누어지며 블록 다이아그램을 (그림 4-1)에 나타내었다.

각 제어부의 기능을 간단히 살펴보면 다음과 같다.

영상획득부는 입력되는 NTSC신호를 A/D변환하여 프레임 메모리부(frame memory)에 저장한다. 주제어부는 영상획득부로부터 프레임 메모리부에 데이터의 저장이 끝났음을 알리는 인터럽트 신호 IRQ2'가 전달되어 오면 저장된 데이터를 DSP의 내부 메모리로 읽어들여 부호화를 행한 후 부호화된 프랙탈계수를 코드 메모리부(coded memory)에 저장한다. 출력제어부는 주제어부로부터 인터럽트 신호 IRQ2'을 받으면 코드 메모리부에 저장된 프랙탈 계수를 읽어들여 이를 복호화기로 전송한다. 이외에도 주제어부는 영상획득부에 초당 샘플링 프레임 수와 획득할 영상의 크기를 정해주는 역할을 한다. 주제어부와 출력제어부는 각각의 프로세서를 사용하여 공유메모리인 코드 메모리부를 중심으로 독립적인 작업을 수행하는 MIMD구조로 이루어지며 전체 시스템의 성능을 향상시키기 위하여 영상획득부, 주제어부, 출력제어부로 이루어지는 3단계의 파이프라인으로 동작하도록 하였다.



(그림 4-1) 제작된 동영상 부호화기의 블록 다이아그램



(그림 4-2) 동영상 부호화기의 파이프라인 동작

〈표 4-1〉 동영상 부호화기의 하드웨어 사양

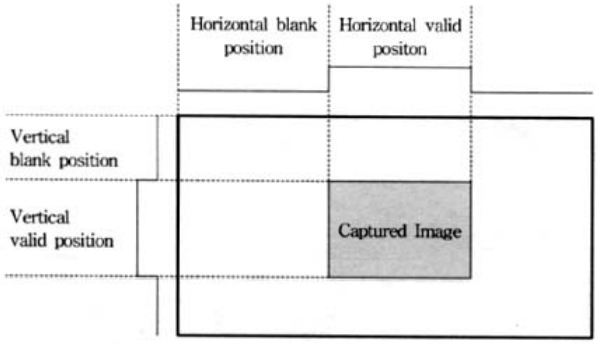
DSP	ADSP2181 (Main & Output controller)
Main Memory	Internal RAM (48kbytes) EEPROM (64kbytes)
Frame Memory	UM611024AK-20 : SRAM (×3 = 384kbytes)
Coded Memory	UM611024AK-20 : SRAM (×2 = 256kbytes)
ADC	Bt252 : 30MSPS (×1)
Output channel	RS422 : Max 1Mbps (×1)
Input channel	NTSC : Max 4 (×1)
Serial com.port	RS232 (×1)

제작된 부호화기에서의 파이프라인 동작은 (그림 4-2)에서와 같다. 파이프라인 동작이 정상적으로 이루어지기 위해서는 주제어부의 부호화 작업(image compress)과 출력제어부의 전송 작업(code transfer)이 영상획득부가 영상 정보를 획득(image grabbing)하는 시간 내에 이루어져야 한다. 부호화 작업이나 전송 작업에 많은 시간이 걸릴 경우 초당 샘플링되는 프레임의 수를 줄이면 각 단계에 할당되는 시간을 늘일 수 있다.

제작된 동영상 부호화기의 하드웨어 사양은 <표 4-1>에 나타내었다.

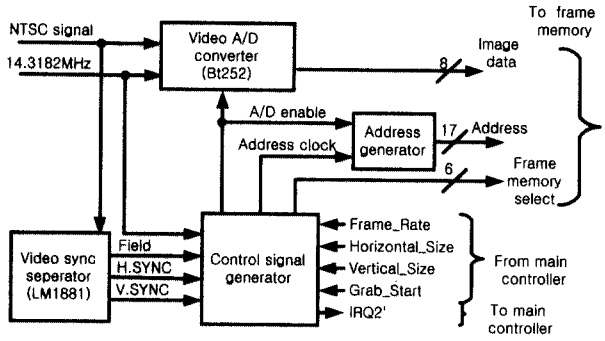
4.1.1 영상획득부

영상획득부는 카메라에서 입력되는 NTSC신호로부터 A/D 변환하여 획득된 디지탈 영상 데이터를 프레임 메모리부에 저장하는 역할을 한다. 초당 A/D 변환하는 프레임의 수와 획득하는 영상의 크기는 주제어부에서 소프트웨어적으로 변화시킬 수 있도록 설계하였다. 한 프레임에서 획득되는 영상의 크기는 (그림 4-3)에서와 같다. 무효 위치(blank position) 수만큼의 영상 정보는 무시하고 유효 위치(valid position) 수만큼의 영상 정보를 A/D 변환하여 저장한다. 즉, 영상의 시작위치는 무효 위치의 수에 의해 결정되는데 DIP스위치를 이용하여 크기를 변화시킬 수 있도록 하였다.



(그림 4-3) 한 프레임 영상에서 획득된 영상

영상획득부의 블록 다이아그램은 (그림 4-4)에서와 같다. 동기 신호 분리단(video sync. seperator)은 입력되는 NTSC 신호로부터 동기 신호를 분리해서 제어 신호 발생단(control signal generator)으로 전달한다. 제어 신호 발생단은 주제어부로부터 입력되는 초당 샘플링 프레임 수와 영상의 크기에 해당하는 시간동안 A/D 변환 가능신호를 A/D 변환단(A/D converter)으로 전달하고 프레임 메모리부에 대한 제어신호와 어드레스를 발생시킨다.



(그림 4-4) 영상획득부의 블럭 다이아그램

4.1.1.1 동기신호 분리단

NTSC신호는 각 프레임마다 동기를 맞추기 위해 수직 동기신호, 수평 동기신호, 필드(field) 신호를 사용한다. NTSC 신호에서는 한 프레임을 나타내기 위하여 두 개의 필드를 사용하는데 필드가 시작됨을 알리는 것이 수직동기신호이며 각 필드에서 한 수평라인의 시작을 나타내는 것이 수평동기 신호이다. 동기신호 분리단에서는 LM1881[18]을 이용하여 NTSC신호로부터 이러한 동기신호를 얻는다.

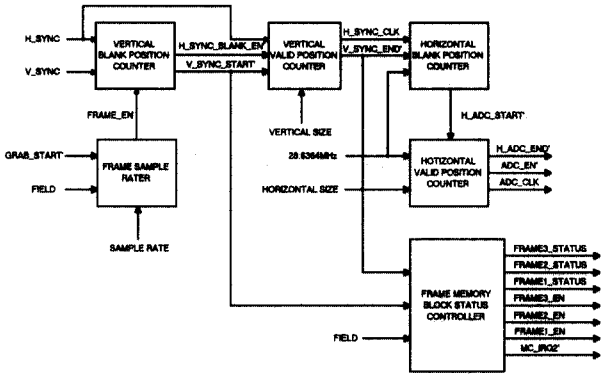
4.1.1.2 제어신호 발생단

제어신호 발생단은 주제어부로부터 입력되는 초당 샘플링 프레임수 제어신호와 영상의 크기에 해당하는 화소 변환 시간동안 A/D 변환 가능신호를 A/D 변환단(A/D converter)으로 전달하고 프레임 메모리부에 대한 제어신호와 어드레스 생성을 위한 클럭을 발생시킨다.

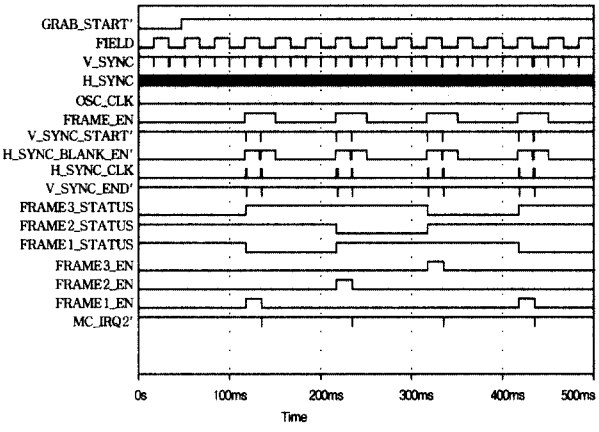
제어신호 발생단의 블록다이아그램을 (그림 4-5)에 나타내었다. 프레임 샘플 레이트(frame sample rate)는 초당 샘플 프레임 수에 맞도록 필드신호를 계수하여 해당하는 프레임을 찾는다. 수직 위치계수기는 필드 내에서 한 수평라인의 시작을 알리는 수평 동기신호를 계수하여 해당하는 수직 유효위치를 찾는다. 수평 위치계수기는 A/D 변환기의 샘플링 주파수를 계수하여 수평 라인내에서 해당하는 수평 유효위치를 찾는다. 프레임 메모리부 블록 상태 제어기는 영상획득부가 어느 프레임 메모리부 블록을 사용하고 있는지를 나타내어 주제어부가 같은 메모리 블록에 접근하지 않도록 설계하였다.

프레임 샘플 레이트는 초당 10프레임, 수직 무효위치와 수직 유효위치는 각각 15일 경우에 수평 위치 계수기를 제외한 제어신호 발생단을 MicroSim에서 시뮬레이션한 결과 파형은 (그림 4-6)에서와 같다. FRAME1_EN, FRAME2_EN,

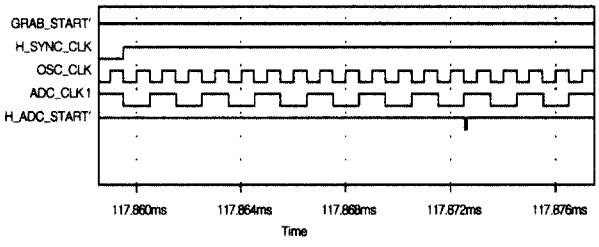
FRAME3_EN은 프레임 메모리부 블록에 대한 제어신호로 사용되며, FRAME1_STATUS, FRAME2_STATUS, FRAME3_STATUS는 각 프레임 메모리부 블록의 사용 상태를 주제어부에 알려주는 역할을 한다.



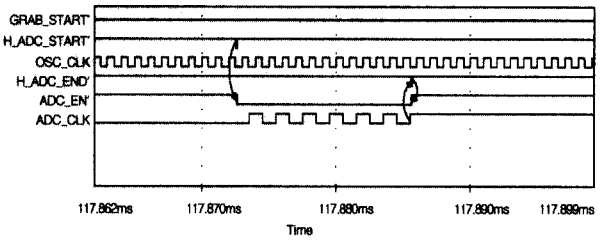
(그림 4-5) 제어신호 발생단의 블록 다이아그램



(그림 4-6) 제어신호 발생단의 시뮬레이션 결과



(그림 4-7) 수평 무효위치 계수기의 시뮬레이션 결과



(그림 4-8) 수평 유효위치 계수기의 시뮬레이션 결과

수평 유효위치와 수평 무효위치가 각각 7일 경우의 시뮬레이션 결과는 (그림 4-7)와 (그림 4-8)에서와 같다. ADC_EN'신호는 A/D 변환단과 어드레스 발생단의 A/D 변환 가능 신호로 사용되며, ADC_CLK는 어드레스 생성단의 어드레스 클럭으로 사용된다.

4.1.1.3 A/D 변환단

A/D 변환단에서는 영상 데이터를 디지털로 변환하기 위하여 Bt252[19]를 이용하였다. Bt252는 8비트의 흑백 데이터를 출력하며 최대 샘플수는 30 MSPS(Mega Samples Per Second)이다. 내부에 룩업 테이블이 내장되어 있어 아날로그 입력에 해당하는 출력 데이터를 조절할 수 있다. 제작된 부호화기에서는 14.3182 MSPS로 A/D 변환을 수행하였으며 최소 입력전압부터 최대 입력전압까지 등간격을 가지는 선형 룩업 테이블(linear look-up table)을 사용하였다.

4.1.1.4 어드레스 발생단

어드레스 발생단은 제어신호 발생부에서 입력되는 어드레스 클럭을 이용하여 A/D 변환된 데이터를 저장하기 위하여 프레임 메모리부의 어드레스를 발생시킨다. 프레임 메모리부의 각 블록은 128KBytes의 어드레스 공간을 가지므로 17비트 동기식 카운터를 이용하여 어드레스를 발생시킨다.

4.1.2 프레임 메모리부

프레임 메모리부는 영상 데이터를 저장하는 곳으로 영상 획득부가 데이터를 저장하고 주제어부가 동영상 부호화를 위하여 데이터를 읽어 가는 구조로 되어있다. 제작된 부호화기에서는 영상획득부와 주제어부의 파이프라인 동작으로 인해 영상 획득과 동영상 부호화가 동시에 이루어지므로 정상적인 파이프라인 동작을 위하여 프레임 메모리부는 세

〈표 4-2〉 프레임 메모리부 블록의 상태

FRAME 1 _EN	FRAME 2 _EN	FRAME 3 _EN	FRAME 1 _STATUS	FRAME 2 _STATUS	FRAME 3 _STATUS	Status
H	L	L	L	H	H	Image grabber is saving data to frame memory block 1.
L	L	L	L	H	H	Image grabber finished saving data to frame memory block 1.
L	H	L	H	L	H	Image grabber is saving data to frame memory block 2.
L	L	L	H	L	H	Image grabber finished saving data to frame memory block 2.
L	L	H	H	H	L	Image grabber is saving data to frame memory block 3.
L	L	L	H	H	L	Image grabber finished saving data to frame memory block 3.

개의 메모리 블록으로 이루어진다.

영상획득부는 A/D 변환된 데이터를 저장하기 위하여 한 개의 블록을 사용하며 주제어부는 동영상 부호화를 위하여 두 개의 블록을 사용한다.

영상획득부가 프레임 메모리부를 사용할 때 각 프레임 메모리부 블록의 상태는 <표 4-2>에서와 같다. 프레임 메모리부 블록에서 데이터의 충돌을 피하기 위하여 주제어부는 동영상 부호화를 위해 데이터를 읽어가기 전에 각 메모리 블록의 상태를 확인한 후 가장 최근에 저장이 끝난 블록의 데이터를 현재 프레임의 데이터로 읽어간다.

4.1.3 주제어부

주제어부는 영상획득부를 초기화시키고 동영상 부호화를 수행하는 역할을 하며 Analog Devices사의 ADSP2181을 사용하였다. ADSP2181은 ALU(Arithmetic and Logic Unit), MAC(Multiplier and ACcumulator), Barrel shifter 등의 기능 블록을 가지고 있으며 내장된 프로그램 메모리, 데이터 메모리와 인스트럭션 파이프라인을 이용하여 CPI(Clock Per Instruction)가 0.5인 특징이 있다. 또한 큰 용량의 데이터는 DMA(Direct Memory Access)를 통하여 억세스하는 바이트 메모리(Byte memory)를 사용한다.

제작된 부호화기에서 ADSP2181은 16.67MHz의 주클럭을 이용하여 33MIPS의 속도로 동작하며, 프레임 메모리부와 코드 메모리부는 바이트 메모리 공간에 할당하였다. 이들 메모리는 칩 내부의 데이터 메모리에 비해 4개의 웨이트 스테이트(wait state)를 가지므로 이에 의한 속도의 지연이 크다.

주제어부의 메모리 맵은 <표 4-3>이며 I/O 맵은 <표 4-4>와 같다. <표 4-4>에서 0x00~0x0A는 영상획득부의 제어를 위해 할당되었으며 0x80~0x85는 프레임 메모리부와 코드 메모리부에서 충돌이 일어나지 않도록 프레임 메모리부와 코드 메모리부의 상태를 살피기 위해 사용한다. 0x86은 주제어부에서 부호화가 끝났음을 출력제어부에 알리는 인터럽트를 발생시키기 위해 사용한다.

〈표 4-3〉 주제어부의 메모리 맵

Program memory		Data memory		Byte memory	
Address	Memory	Address	Memory	Address	Memory
0x0000 ~ 0x3FFF	Internal program memory	0x0000 ~0x3FDF	Internal data memory	0page ~3page	Booting ROM (64K bytes)
				8page ~15page	Frame memory 1 (128 KBytes)
				16page ~23page	Frame memory 2 (128 KBytest)
		0x3FE0 ~0x3FFF	Memory mapped register	24page ~31page	Frame memory 3 (128 KBytes)
				32page ~39page	Common memory 1 (128 KBytes)
				40page ~47page	Common memory 2 (128 KBytes)

〈표 4-4〉 주제어부의 I/O 맵

Address	I/O signal name
0x00	GRAB_START'
0x01	FRAME_SAMPLE'
0x02	V_SIZE'
0x03	H_SIZE'
0x04	GRAB_END'
0x08	ADC_ADDRREG'
0x09	ADC_RAMDATA'
0x0A	ADC_COMREG'
0x80	FM_STATUS_RD'
0x81	FM_IMAGE_RD'
0x84	CM_STATUS_WE'
0x85	CM_STATUS_RD'
0x86	OC_IRQ2'

4.1.4 코드 메모리부

주제어부에 의해 동영상 부호화가 끝나면 부호화된 데이터를 코드 메모리부에 저장하고 출력제어부는 저장된 데이터를 읽어와서 복호화기로 전송하게 된다. 코드 메모리부는 두 개의 블록으로 이루어지며 주제어부가 한 개, 출력제어부가 한 개의 블록을 동시에 사용한다. 두 제어부가 동시에 같은 메모리 블록을 억세스함으로써 발생하는 버스의 충돌을 방지하기 위하여 각 제어부는 먼저 메모리 블록의 상태를 살펴서 사용하지 않고 있을 경우에만 억세스를 시작하도록 프로그래밍 하였다.

4.1.5 출력제어부

출력제어부는 부호화된 데이터를 복호화기로 전송하는 역할을 하며 주제어부와 마찬가지로 ADSP2181을 사용하였다. 출력제어부의 I/O 맵과 메모리 맵을 <표 4-5>와 <표 4-6>에 나타내었다. <표 4-6>에서 0x10은 RS422 출력 포트를 제어하기 위해 사용된다.

〈표 4-5〉 출력제어부의 I/O 맵

Address	I/O port
0x10	DUART_CS'
0x20	OC_CM_STATUS_WE'
0x21	OC_CM_STATUS_RD'
0x22	MC_IRQE'

〈표 4-6〉 출력제어부의 메모리 맵

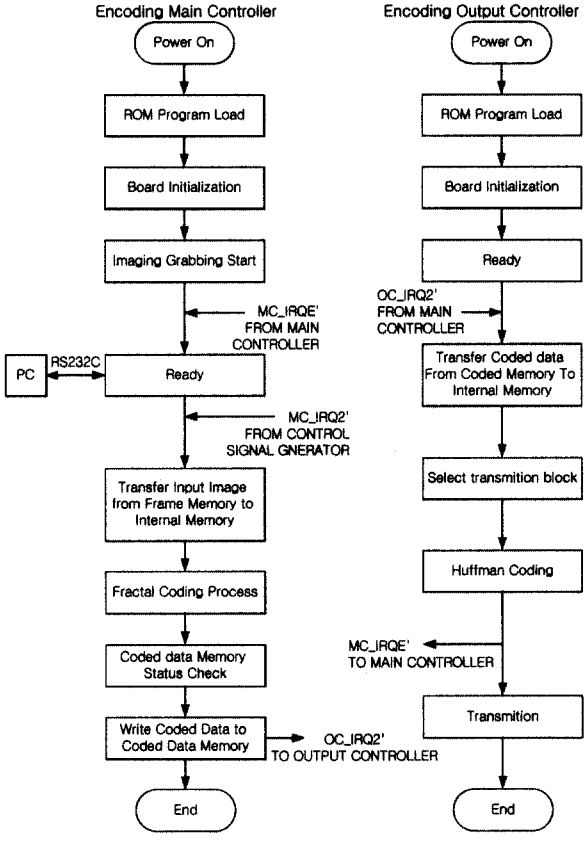
Program memory		Data memory		Byte memory	
Address	Memory	Address	Memory	Address	Memory
0x0000 ~ 0x3FFF	Internal program memory	0x0000 ~ 0x3FDF	Internal data memory	0page ~3page	Booting ROM (64K bytes)
				8page ~15page	Coded memory 1 (128 KBytes)
		0x3FE0 ~ 0x3FFF	Memory mapped register	16page ~23page	Coded memory 2 (128 KBytest)

4.2 동영상 부호화기의 소프트웨어

제작된 부호화기의 모든 프로그램은 C 크로스-컴파일러를 사용할 경우에 발생할 수 있는 코드수의 증가와 이에 따른 속도의 저하를 피하기 위하여 어셈블리어(assembly language)로 작성하였다. 또한 부호화 연산에 필요한 부동 소수점은 ADSP2181의 기능 블록에서 사용하는 고정 소수점 형식과 호환되도록 사용자 정의 부동 소수점 형식을 사용하였다.

4.2.1 프로그램의 흐름도

제작된 부호화기에서는 파이프라인을 사용하므로 주제어부의 부호화와 출력 제어부의 전송 작업은 이전 단계의 작업이 끝나면 인터럽트에 의해 기동되도록 함으로써 파이프라인이 정상적으로 동작하도록 프로그래밍 하였다. 제작된 부호화기의 프로그램 흐름도는 (그림 4-9)에서와 같다.



(a) 주제어부의 흐름도 (b) 출력제어부의 흐름도

(그림 4-9) 프로그램의 흐름도

4.2.2 부동소수점 형식

ADSP2181에서는 소수점 연산을 위하여 고정소수점 형식인 1.15 형식을 사용하며 IEEE-754 floating-point standards[20]에서 사용하는 히든 비트(hidden bit)는 사용하지 않는다. 1.15 형식은 MSB가 사인 비트(sign bit)로 사용되며 나머지 15 비트는 소수 비트(fractional bit)이다. 따라서 1.15 형식에서 나타낼 수 있는 수의 범위는

$$-1.0 \leq \text{significand} < 1.0$$

같다.

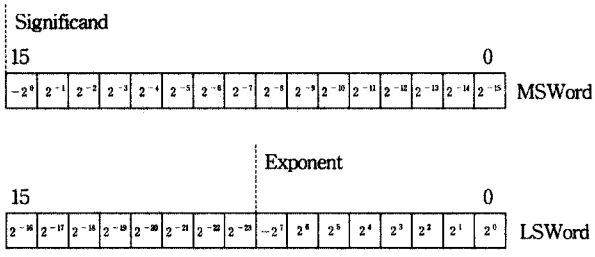
ADSP2181의 경우 부동소수점 연산을 지원하지 않지만, 주제어부의 프랙탈 부호화 프로그램에서는 부동소수점 연산이 필요하다. 부동소수점을 사용하는 프로그램을 작성할 때 ADSP2181의 내부 연산과 호환을 유지하기 위하여 히든 비트를 사용하지 않는 사용자 정의 부동소수점 형식을 만들어 사용하였다. 사용자 정의 부동소수점에서의 시그니피컨드(significand)는 자리 수를 표시하는 사인 비트(redundant sign bits)를 제거한 1.23 형식을 이용하였다. 그리고 부동소수점의 익스포넌트(exponent) 값은 8비트로 할당하였으며 2의 보수를 사용하므로 -128~+127의 범위를 갖는다.

제작된 부호화기에 사용된 사용자 정의 부동소수점 형식에서 각 비트의 웨이팅(weighting)은 <표 4-7>에서와 같다.

<표 4-7>에서 LSWord의 시그니피컨드는 부호화 처리 속도를 증가시키기 위하여 연산 결과의 정확성이 요구되지 않는 경우에는 사용하지 않도록 하였다.

2 바이트 정수와 4 바이트 정수를 사용자 정의 부동소수점 형식으로 변환을 위한 프로그램은 (프로그램 4-1)과 같다.

〈표 4-7〉 사용자 정의 부동소수점 형식




```
{ ========================================
  convert integer to floating point format
    mr1  : input
    ar   : exponent of result,      sr1 : significand of result
========================================= }
itof :      se = exp mr1 (HI) ;   { sign extention bit의 수를 구
                                    한다. }
            ax0 = 15 ;
            sr = norm mr1(HI) ;   { sign extention bit가 없도록
                                    left shift. }
            ay0 = se ;
            ar = ax0 + ay0 ;      { new exponent }
            rts ;

{ ================================================
  convert long integer to floating point format
    mr1  : MSW of input,          mr0 : LSW of input
    ar   : exponet of resultsr1,  sr0 : significand of result
================================================= }
longtof :   se = exp mr1 (HI) ;
            se = exp mr0 (LO) ;
            ax0 = 31 ;
            ay0 = se ;
            ar = ax0 + ay0 ; { compute exponent }
```

```
ar = ax0 + ay0 ; { compute exponent }
sr = norm mr1 (HI) ;
sr = sr or norm mr0 (LO) ; { normalize significand }
rts ;
```

(프로그램 4-1) 정수의 부동소수점 변환 프로그램

5. 실험 및 결과

제작된 프랙탈 영상 부호화기의 성능을 추정하기 위하여 QCIF 포멧(144×176)의 Claire, Miss America 그리고 Alex 영상 100 프레임을 사용하여 부호화를 수행하였다. 제작된 부호화기의 성능 실험은 RS232 직렬 통신을 이용하여 PC의 제어에 의해 이루어졌다. 성능을 평가하기 위한 기준으로는 부호화 시간, PSNR(peak-to-peak signal to noise ratio), 화소당 비트수(BPP)를 사용하였다. <표 5-1>은 부호화에 소요되는 비트 수를 각 계수에 대해 정리한 것이다.

100 프레임의 동영상을 제작된 부호화기에서 4×4 한 종류의 블록을 이용하여 영상 전체를 부호화한 경우와 평탄임계치의 변화에 따른 성능평가를 <표 5-2>에 나타내었다. 이는 정지영상에 대하여 부호화 할 경우에 치역블록 8×8에 대하여 프랙탈 계수 두개와 네 개를 적용하였을 때의 평균 화질, BPP 및 프레임당 평균부호화 시간을 비교하였다. <표 5-2>에서 4×4 블록의 경우와 프랙탈 계수 $a_3^{(k)}$의 값을 수정한 4×4 블록에 대한 프레임당 평균 부호화 시간은 61.46ms와 81.57ms를 갖고 BPP는 1.25로 대체로 높다.

본 논문에서 제작된 부호화기에서 평탄임계치 $TH_S = 64$로 설정하고 부호화하면 4×4 블록으로만 부호화 한 경우와 비교하면 다음과 같다.

〈표 5-1〉 프랙탈 계수에 할당된 비트수

Motion Estimation	Classification Block	Luminant Shift	Gred X	Gred Y	Scaling
1	1	6	5	5	4

Claire 테스트 영상에 대한 프레임 평균화질은 0.14dB 떨어진 반면에 부호화 시간은 18.5%, 비트율은 50% 이상 줄일 수 있었다. 이는 기존의 쿼드-트리 방식[21-23]은 치역블럭과 복원된 치역블럭의 오차가 클 경우에 분할과 큰 블럭에서 프랙탈 계수의 개수를 증가시킴으로 프레임당 블럭의 수를 줄여 비트율을 줄인다. 기존의 쿼드-트리 방식은 비트율에서는 잇점이 있어나, 복호화 시간이 부호화 시간에 포함되어 실시간 부호화기 구현을 어렵게하는 단점이 있다.

본 논문의 프랙탈 영상 부호화는 정의역 탐색을 하지않는 확장된 SAS(Self Affine System)를 이용하기 때문에 부호화시간은 치역블럭의 분할과 프랙탈 계수의 개수에 밀접한 관계를 갖는다. 8×8 치역블럭의 분할을 위하여 치역블럭의 복원된 영상을 이용하지않고 8×8 치역블럭과 공간축소된 8×8 정의역블럭과의 유사성 정도에 따라 블럭 분할이 이루어진다. 그 결과, 8×8 치역블럭으로 부터 네 개의 4×4 치역블럭으로 분할 판정을 위해 소요되는 시간은 전체 부호화 시간을 증가시킨다. 따라서 Wilson[8]등에 의한 알고리즘과 비교하여 볼 때, 8×8 치역블럭이 차지하는 비율이 높을수록 비트율 및 부호화 시간이 감소되는 효과를 갖는다.

그 결과, 제작된 프랙탈 영상 부호화기에서는 QCIF 형태의 Claire 영상 전체를 최소 초당 12프레임의 속도로 부호화가 가능하고, Claire 동영상의 경우는 정지영상의 54%를 부호화를 행하면 초당 30프레임의 속도로 QCIF 형태의 영상을 부호화 가능함을 알 수 있다.

동영상에 대한 제작된 부호화기의 성능 평가는 <표 5-3>에 나타내었다. 부호화할 블록을 선택하기 위해 이전 프레임과 현재 프레임의 움직임 정도를 찾는 움직임 임계치 TH_m와 현재 프레임내에서 정의역 블록과 치역 블록의 유사 정도를 나타내는 평탄 임계치 TH_s의 변화에 따른 부호화 성능 평가를 위하여 100프레임의 평균 성능을 나타낸다.

<표 5-3>는 (그림 3-1)에서 제안된 동영상의 프레임간 중복성을 고려한 제작된 부호화기의 성능평가을 나타낸다. 부호화할 블럭을 선택하기 위해 이전 프레임과 현재 프레임의 움직임 정도를 찾는 움직임 임계치와 현재 프레임내에서 정의역블럭과 치역블럭의 유사 정도를 나타내는 평탄 임계치의 변화에 따른 부호화 성능 평가를 나타낸다.

본 논문의 부호화기를 통하여 움직임이 있는 블럭중에서 $THS = 225$이고 $TH_m = 5 \sim 15$ 사이로 변화되면 Clarie 영상에서는 움직임이 있는 영역의 1.5%~2.8%, Miss America 영상에서는 12.3%~24.0%, 그리고 Alex 영상에서는 8.4%~9.8%가 8×8 치역블럭으로 판정되었다.

〈표 5-2〉 제작된 부호화기를 이용한 정지영상에 대한 성능 평가

Block Partition		Claire			Miss America			Alex		
		PSNR	BPP	T(ms)	PSNR	BPP	T(ms)	PSNR	BPP	T(ms)
4×4[5]		33.61	1.250	81.57	36.92	1.250	86.09	35.23	1.250	96.69
8×8, 4×4. Modified	$TH_s = 64$	33.47	0.610	66.44	35.50	0.705	72.14	34.77	0.978	89,09
	$TH_s = 100$	33.35	0.586	65.31	35.26	0.647	69.42	34.71	0.944	85.24
	$TH_s = 150$	33.24	0.568	64.31	34.53	0.517	64.76	34.02	0.860	88.78
	$TH_s = 400$	32.74	0.507	61.02	33.89	0.451	60.36	32.92	0.779	77.69

〈표 5-3〉 제작된 부호화기를 사용한 동영상에 대한 성능 평가

Threshold Value \ Image		Claire			Miss America			Alex		
TH_m	TH_s	PSNR	BPP	T(ms)	PSNR	BPP	T(ms)	PSNR	BPP	T(ms)
5	64	32.89	0.159	25.64	33.61	0.252	27.19	34.72	0.180	32.87
	100	32.83	0.158	25.60	33.55	0.245	27.38	34.70	0.179	32.64
	225	32.77	0.156	25.58	33.27	0.221	26.98	34.27	0.170	32.07
	400	32.61	0.150	25.40	32.89	0.202	26.60	33.60	0.156	31.57
10	64	32.48	0.122	22.90	32.30	0.191	22.61	34.25	0.114	28.13
	100	32.44	0.121	22.89	32.27	0.187	22.38	34.23	0.113	28.02
	225	32.40	0.120	22.85	32.12	0.172	22.26	33.88	0.107	27.69
	400	32.31	0.116	22.71	31.86	0.160	22.00	33.39	0.098	27.36
15	64	32.17	0.102	21.46	31.29	0.160	20.93	33.78	0.094	25.85
	100	32.13	0.102	21.45	31.27	0.158	20.93	33.77	0.094	25.79
	225	32.11	0.101	21.45	31.16	0.148	20.82	33.56	0.089	25.56
	400	32.04	0.098	21.35	30.98	0.139	20.65	33.09	0.082	25.33

본 논문에서 제작된 부호화기는 프랙탈 부호화 계수의 개수 이외에 치역블럭의 전송유무 또는 치역블럭 어드레스를 나타내는 비트가 상대적으로 줄어든다. 블럭분할여부에 할당된 비트수를 포함하여도 줄어든 비트수는 일정한 전송율에서 프랙탈 부호화 계수 비트의 비율을 증가시키는 잇점이 있었다.

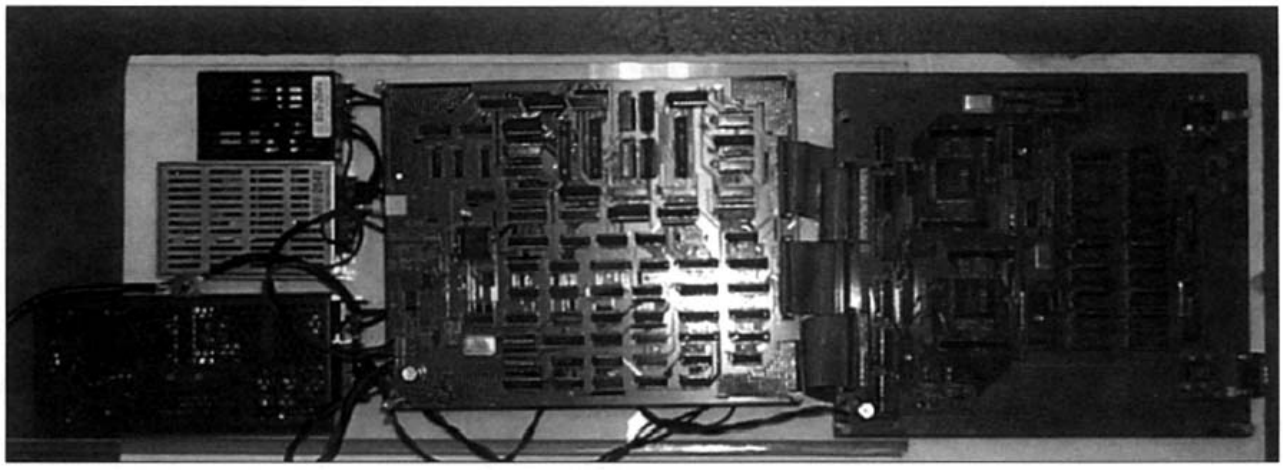
제작된 부호화기에서는 움직임 및 평탄임계치들의 값에 따라 복원 영상화질, 비트율 및 부호화 시간은 가변적이다. <표 5-3>에서 $TH_m = 10$, $TH_S = 64$ 일 때, 평균 부호화 시간은 Claire, Miss America 영상에 대하여 25.64ms, 27.19ms, 32.87ms 소요됨으로 초당 30프레임이상의 속도로 실시간 부호화 처리가 가능하였다.

(그림 5-1)는 제작된 동영상 부호화기에서 초당 샘플 레이트가 30프레임일 때 획득한 연속된 3프레임의 영상을 PC로 다운로드 받은 것이다. (그림 5-2)은 제작된 동영상 부호화기에서 부호화된 데이터를 PC로 다운로드 받은 후 PC에서 복원한 영상이다. (그림 5-3)는 제작된 동영상 부호화기의 사진이다.

(a) 첫 번째 프레임 (b) 두 번째 프레임

(c) 세 번째 프레임

(그림 5-1) 제작된 동영상 부호화기에서 획득한 연속 3프레임의 영상



(그림 5-3) 제작된 동영상 부호화기의 사진

(a) 두 번째 프레임 (b) 세 번째 프레임

(그림 5-2) 제작된 부호화기에서 부호화된 데이터를 PC에서 복원한 영상

6. 결 론

본 논문에서는 프랙탈 이론을 이용한 영상 부호화 알고리즘을 탑재한 실시간으로 부호화 할 수 있는 부호화기를 범용 DSP 프로세서들로 사용하여 프랙탈 알고리즘 기반의 영상신호 부호화기를 제작하고 그 성능을 평가하였다.

본 논문에서 제작된 부호화기는 2개의 ADSP2181을 이용하여 3단계의 파이프라인 구조에 의하여 영상신호의 부호화과정을 수행한다. 첫 번째 파이프라인 과정은 NTSC 표준영상신호의 AD변환을 담당하고 두 번째 과정은 프레임 메모리에 저장된 영상데이터를 제안된 알고리즘에 의하여 프랙탈 변환을 수행한다. 세 번째 과정은 프랙탈 부호화된 코드에 Huffman coding을 수행하여 RS422포트에 출력한다.

제작된 부호화기의 성능테스트 결과 QCIF 포맷의 영상데이터에 대하여 정지영상에 대하여 부호화 할 경우 Claire, Miss America, Alex 영상에 대하여 66.44㎳, 72.12㎳, 89.09㎳의 부호화 속도를 얻었으며 화면간 및 화면내의 중복성을 이용하여 복호화 할 경우 데스트한 Claire, Miss America, Alex등의 영상에 대하여 움직임 임계치 $TH_m = 5$, 분할임계치 $TH_s = 64$일 때 25.64㎳, 27.19㎳, 32.87㎳의 프레임당 평균 부호화 속도를 얻었다. 그 결과, 실험 영상 모두에 대하여 초당 평균 30 프레임 이상의 부호화 속도를 얻어 실시간 처리가 가능하였다.

참 고 문 헌

[1] M. F. Barnsley and Lyman P. Hurd, *Fractal Image Compression*. AK Prters, Ltd.

[2] A. E. Jacquin, "A novel fractal block-coding technique for digital images," *Proc. IEEE ICASSP*, pp.2225-2228, 1990.

[3] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformation," *IEEE Trans. on Image Processing*, Vol.1, No.1, pp.18-30, Jan., 1992.

[4] S. Lepsoy, G. E. Oien and T. A. Ramstad, "Attractor image compression with a fast non-iterative decoding algorithm," *Proc. IEEE ICASSP '93*, Vol.5, pp.337-340, Apr., 1993.

[5] D. M. Monro and F. Dudbridge, "Fractal approximation of image blocks," *Proc. IEEE ICASSP '92*, Vol.3, pp.485-488, 1992.

[6] B. Hurtgen and P. Buttgen, "Fractal approach to low rate video coding," *Proc. SPIE VICP '93*, Vol.2094, pp.120-131, 1993.

[7] B. Hurtgen and C. Stiller, "Fast hierarchical codebook search for fractal coding of still images," *Proc. SPIE VICP '93*, Vol.1977, pp.397-408, 1993.

[8] D. L. Wilson, J. A. Nicholls, and D. M. Monro, "Rate buffered fractal video," *Proc. IEEE ICASSP '94*, pp.505-508, 1994.

[9] 이성훈, 김재철, 박종식, "연속 영상을 위한 프랙탈 부호화", *제8회 신호처리합동학술대회논문집*, Vol.8, No.1, pp.1039-1043, 1995.

[10] 김재철, 박종식, "PC를 이용한 실시간 프랙탈 부호화 구현", *한국통신학회논문지*, 제21권 제11호, pp.2789-2800, 1996.

[11] *ADSP-2100 Family User's Manual*, Analog Devices, Sep., 1995.

[12] *Circuit Analysis User's Guide*, MicroSim Corporation, Apr., 1995.

[13] B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Co., San Francisco, 1982.

[14] Peitgen, Jurgen, and Saupe, *Chaos and Fractal New Frontiers of Science*, Springer Verlag, New York, 1992.

[15] M. F. Barnsley, *Fractals Everywhere*, Academic Press, 1988.

[16] R. L. Burden and J. D. Fairs, *Numerical Analysis third edition*, PWS Publishers, 1985.

[17] *ADSP-2100 Family C Tools Manual*, Analog Devices.

[18] *Special Purpose Linear Devices Databook*, National Semiconductor, 1989.

[19] *Brooktree Product Databook*, Brooktree Corporation, 1989.

[20] David A. Patterson and John L. Henessey, *Computer Organization & Design : The Hardware/Software Interface*, Morgan Caufmann Publishers.

[21] Greg Viens and Monson H. Hayes, III, "Adaptive IFS image coding with proximity maps," *Proc. IEEE ICASSP-93*, pp.349-352, 1993.

[22] 도원, 서정태, 윤대희, "가변블럭적용 프랙탈영상압축 알고리즘", 제7회 신호처리합동학술대회논문집, 제7권 제1호, 1994.

[23] Y. Fisher, *Fractal Image Compression : Theory And Application*, Springer-Verlag, New York, 1994.

김 재 철

e-mail : blackcow@kumi.ac.kr
1990년 경북대학교 전자공학과(학사)
1992년 경북대학교 전자공학과(석사)
2000년 경북대학교 전자공학과(공학박사)
1997년~현재 구미1대학 정보통신전공 조교수
주관심분야 : 실시간 영상처리, 디지털회로설계, 마이크로프로세서 응용

최 인 규

e-mail : powercik@hanmail.net
1995년 경북대학교 전자공학과 졸업(학사)
1997년 경북대학교 전자공학과(석사)
2002년 경북대학교 전자공학과(박사)
2002년~현재 삼성전자 무선사업부
관심분야 : 디지털회로설계, ASIC설계