

# 무곱셈 대각 보간 필터의 설계 및 VLSI 구현에 관한 연구

## (On the Design Technique and VLSI Structure for a Multiplierless Quincuncial Interpolation Filter)

崔 鎮 宇,\* 李 商 郁\*

(Jin Woo Choe and Sang Uk Lee)

### 要 約

영상신호의 이차원 필터링에 필요한 많은 곱셈 때문에 이차원 필터의 실시간 구현은 용이하지 않다. 이에 본 논문에서는 효과적인 무곱셈 필터의 필터 설계기법과 VLSI 구조가 제시되었다. 기술된 설계기법은 필터 이차원 주파수응답의 효과적인 이산화를 통해 요구되는 계산량을 기존의 기법에 비해 절감하고 있다. 실험을 통해 5×5 필터의 경우 기존의 기법에 비해 기술된 설계기법의 소요시간이 약 80%정도 단축될 수 있으며 설계된 필터의 성능은 근접함을 알 수 있다. 또한 본 논문에는 무곱셈 필터의 구현을 위해 두가지 다른 VLSI 구조가 제안되었다. 두가지 구조는 각각 블록단위 병렬처리와 라인단위 병렬처리를 전제로 하고있으며 두 구조 모두에 있어  $W_u$ 의 연속계수 필터의 구조에 비해 향상된 AP 값을 보인다.<sup>[4]</sup>

### Abstract

A huge amount of multiplications is required for 2-D filtering on the image data, making it difficult to implement a real-time quincuncial interpolator. In this paper, efficient design technique and VLSI structures for 2-D multiplierless filter are presented. In the filter design, by introducing an efficient scheme for discretizing the frequency response of the prototype filter, it is shown that a significant amount of the computational burden required in the conventional techniques, such as local search, branch and bound techniques, could be saved. In the case of 5×5 filter, it is found that the design technique described in this paper could save about 80% of the computation time, compared to the conventional methods, while providing a comparable performance. For a hardware implementation, two different VLSI structures for 2-D multiplierless filter are also introduced in this paper: One is for block parallel processing and the other for scan-line parallel processing. In both structure, the AP(area-period) figure improves over Wu's structure[4].

### I. 서 론

대각(quincuncial) 표본줄임(subsampling) 및 보간(interpolation)은 영상신호의 압축을 위해 널리 사용되는 방법으로 최근 HD-MAC 방식에서 상용됨에 따라 많

은 관심의 대상이 되고 있다.<sup>[1,10]</sup> 대각 표본줄임 및 보간 시 널리 상요되는 이차원 마름모형의 필터는 분리가능(separable)하지 않으므로 구현 또한 이차원으로 이루어져야 한다. 이차원 필터로는 영상신호처리시 선형 위상특성이 중요하므로 FIR(finite impulse response) 필터가 주로 사용되며  $(2M+1) \times (2M+1)$  이차원 FIR 필터(filter)의 경우 한 출력 화소를 계산하기 위해  $(2M+1)(2M+1)$  번의 곱셈이 필요하다<sup>[2]</sup>. 이러한 많은 곱셈은

\*正會員, 서울대학교 制御計測工學科  
(Dept. of Cont. & Inst. Eng., Seoul Nat'l Univ.)  
接受日字: 1992年 2月 17日

높은 전송율의 영상신호에서는 실시간 구현에 장애가 된다.<sup>[4]</sup> 따라서 계수와 화소값의 곱을 덧셈만으로 구할 수 있는 무곱셈(multiplierless) 필터를 사용하는 것이 실시간 구현에 유리하다.<sup>[5]</sup> 본 논문의 목적은 대각 보간에 적합한 이차원 무곱셈 FIR 필터의 설계기법을 제안하고 실시간 구현을 위한 VLSI 구조를 제시하는데 있다.

우수한 성능의 무곱셈 대각 보간 필터를 설계하기 위해서는 대각 보간에 적합한 필터의 특성을 조사하여야 한다. 즉 최대최소(minimax)의 기준으로 필터를 설계할 때에는 이상적인 필터(ideal filter)의 주파수응답(frequency response), 대역별 가중치(weight), 적절한 차수등이 결정되어야 한다.<sup>[6]</sup> 그러나 대각 보간 필터의 성능은 각 영상신호의 주파수대역(frequency domain) 에너지(energy) 분포에 의존하므로 모든 영상신호에 대해 최적인 이차원 필터는 구할 수 없다. 따라서 가능한 많은 영상신호에 대해 실험하여 전반적으로 우수한 성능을 나타내는 필터를 찾을 필요가 있다. 이에 본 논문에서는 여러 가지 차수와 가중치로 필터를 설계하여 여러 영상신호에 대해 실험함으로써 대각보간에 적합한 필터를 조사 비교하였다.

## II. 이차원 무곱셈 필터의 설계기법

차수가  $(2M+1) \times (2N+1)$ 인 이차원 FIR 필터계수를  $h(m, n)$ 라하면 주파수응답  $H(u, v)$ 는 다음과 같이 표현된다.

$$H(u, v) = \sum_{m=-M}^M \sum_{n=-N}^N h(m, n) e^{-j(mu + nv)} \quad (1)$$

이때 선형위상특성을 위해 필터계수가 원점에 대칭 즉

$$h(m, n) \in \mathbb{R} (m = -M, \dots, 0, \dots, M, n = -N, \dots, 0, \dots, N) \text{ 이고} \\ h(m, n) = h(-m, -n) \quad (2)$$

이라 가정하면 주파수응답  $H(u, v)$ 는 다음과 같이 다시 쓸 수 있다.

$$H(u, v) = \sum_{m=1}^M \sum_{n=1}^N h'(m, n) \cos(mu + nv) \\ + \sum_{n=0}^N h'(0, n) \cos(nv), \quad (3)$$

$$\text{여기서 } h'(m, n) = \begin{cases} h(m, n) & m=0, n=0 \\ 2h(m, n) & \text{otherwise.} \end{cases} \quad (4)$$

일반적으로 설계하고자하는 필터의 이상적 주파수응답은 주파수영역을 여러 대역으로 분할하여 각 대역에서 고정된 값을 요구한다.<sup>[6]</sup> 따라서 FIR 필터로는 이상적 주파수응답을 얻을 수 없고 이에 근사한 주파수응답을 얻도록 필터계수를 최적화하여야한다. 근사의 방법에는 이상적 주파수응답과 실제의 주파수응답의 제곱오차합을 최소화하도록 하는 바업과 최대오차를 최소화하는 방법이 주로 사용되나<sup>[15]</sup>, 본 논문에서는 최적의 대각보간 FIR 필터를 설계하기 위해 최대최소의 기준을 사용한다. 이때 필터성능의 비교 기준이되는 가중최대오차는 다음식으로 표현된다.

$$\text{Maximum error} = \sup_{(u,v) \in P} \{ W(u,v) |H(u,v) - D(u,v)| \} \quad (5)$$

여기서  $H(u, v)$ 는 설계된 필터의 주파수응답을,  $D(u, v)$ 는 설계하고자하는 필터의 이상적 주파수응답을,  $W(u, v)$ 는 주파수대역에서의 오차 가중치를 각각 나타내며  $P$ 는 이차원 주파수영역에서 밀집하게 추출된 점으로 필터의 차수에 따라 집합  $P$ 의 원소수는 조정되어야 한다.<sup>[6]</sup> 집합  $P$ 와 필터차수의 관계는 뒤에 자세히 다루기로 한다.

FIR 필터의 구현식은 일반적으로 컨벌루션으로 나타낼 수 있고 한 출력 최소값을 구하기 위해  $(2M+1)(2N+1)$ 번의 곱셈과  $(2M+1)(2N+1)-1$ 번의 덧셈이 필요하다. 그러나 곱셈을 작은 수의 덧셈으로 가능하게 하는 CSD(canonic signed digit)를 사용하여 필터를 설계함으로써 실시간구현을 용이하게 할 수 있다.<sup>[18]</sup> CSD란 이진표기시 정해진 수의 1 또는 -로 표현되는 수로 다음과 같이 나타낼 수 있다.

$$x = \sum_{k=1}^L s_k 2^{-pk} \quad (6)$$

$$s_k \in \{-1, 0, 1\} \\ p_k \in \{0, 1, 2, \dots, \ell-1\}.$$

여기서  $\ell$ 은 CSD의 전체 자리수를 나타내며  $L$ 은 전체 자리수중 0 이외의 값을 갖을 수 있는 자리의 수를 나타낸다. 따라서  $(L-1)$ 개의 덧셈으로 다른 수와의 곱을 구할 수 있다.  $L < \ell$  일때 수직선상의 CSD의 분포는 매우 불균일하나 0 부근에 많은 값이 집중되어 있다<sup>[5]</sup>. 또한 일반적으로 필터계수는 작은 값을 갖는 것이 대부분이므로 CSD에 적당한 스케일링을 하면 필터계수를 양자화할때 양자화 오차를 감소할 수 있다<sup>[5]</sup>. 따라서 지금까지 CSD 계수 FIR 필터를 설계하는 기법이 다수 소개되었으나 크게 local search 기법과 MILP(mixed integer linear programming)의 일종인 branch and bound 기법의 두

가지 부류로 나눌 수 있다.<sup>[7,8,9]</sup>

### 1. Local search 기법

Local search 기법은 준최적 CSD계수 필터설계기법으로 주로 일차원 필터 설계에 이용한 예가 많으나 이차원 필터설계에도 확장할 수 있다.<sup>[8,9]</sup> 이 설계기법의 원리를 설명하면 다음과 같다. 우선  $\ell$ 과  $L$ 이 결정되면 CSD의 개수  $p$ 도 결정된 즉  $p$ 는  $\ell, L$ 의 함수이다. 설계하려는 CSD계수 이차원 필터의 차수를  $(2M+1) \times (2N+1)$ 이라 하면 식 (2)에 의해 실제 독립적인 필터계수의 수는  $(2N+1)M+1+1$ 개가 된다. 따라서  $p$ 개의 CSD계수를 필터계수  $h(m,n)$ 에 할당하여 설계할 수 있는 서로 다른 CSD계수 필터의 갯수는  $p^{(2N+1)M+N+1}$ 개가 된다. 이는 현실적으로 매우 큰 값으로 이 모든 필터에 대해서 식 (5)을 이용 최대오차를 구해 성능을 비교하는 것은 불가능하다. 이에 local search 기법은 주어진 CSD계수 필터에 이웃하는 보다 좋은 성능의 CSD계수 필터를 탐색하는 과정을 반복하여 준최적 CSD계수 필터를 구한다.

Local search 기법의 장점은 branch and bound 기법에 비해 계산량이 짧다는 것에 있다.<sup>[9]</sup> 또한 branch and bound 기법의 필터차수 증가에 따른 계산량의 증가에 비해 local search 기법은 필터차수의 증가에 따른 계산량의 증가가 작다. Kodek<sup>[9]</sup>에 의하면 현실적으로 branch and bound 기법으로 설계가 불가능한 필터차수에 대해 local search 기법을 사용하여도 필터차수 증가에 따른 필터성능의 향상은 만족할만한 것으로 알려져 있다.

### 2. Branch and bound 기법

Local search 기법은  $\ell, L$ 과 필터의 차수가 주어졌을 때  $P^{(2N+1)M+N+1}$ 개의 가능한 CSD계수 필터중 일부분을 탐색하여 준최적의 CSD계수 이차원 필터를 찾는 기법이다. Branch and bound 기법은 독립적인 필터계수를 하나씩 순차적으로 CSD로 고정하여 최적의 CSD계수 필터를 찾는 기법이다.<sup>[7,9]</sup> 앞서 설명하였듯이  $(2M+1) \times (2N+1)$ 의 이차원 FIR 필터는  $(2N+1)M+N+1$ 개의 독립적인 계수가 존재한다. 이들중 일부 필터계수를 CSD로 고정한 상태에서  $L.P.$ (linear programming)나 ascent 알고리즘을 이용하여 고정되지 않은 연속계수의 최적값과 이때의 최대오차를 구할 수 있다.<sup>[20,21]</sup> 이렇게 구한 최대오차는 CSD로 고정되지 않은 필터계수를 CSD로 고정하여 얻을 수 있는 최대오차의 하한이 된다. 따라서 일부 필터계수를 특정 CSD로 고정하여 얻은 최대오차가 임의의 CSD계수 필터의 최대오차보다 큰 값을 갖을 경우 고정되지 않은 필터계수를 어떤 CSD로 할당하여도 최적의 CSD계수 필터는 얻을 수 없음을 알 수 있다. 이에 branch and bound 기법에서는 필터계수를 순차적으로

여러 CSD로 고정하며 일부 필터계수가 CSD로 고정된 상태에서의 최대오차를 구해 이 최대오차가 임의의 CSD계수 필터의 최대오차보다 크면 더 이상 고정되지 않은 필터계수를 고정하지 않음으로써 필터의 탐색범위를 축소한다.

Branch and bound 기법은 이론적으로 최적의 CSD계수 FIR 필터를 보장한다. 그러나 필터의 차수가 증가함에 따라 분기의 수도 증가하고, 또 집합  $U$ 도  $u, v \in [-n, +n]$ 에서 더욱 밀집해져야 하므로  $L.P.$ 의 제약조건이 증가해 계산량이 급격히 증가한다.<sup>[16]</sup> 따라서 현실적으로 설계가능한 필터차수에 제약이 따르면 이차원 필터의 경우  $L.P.$ 의 제약조건이 증가해 설계 가능한 필터의 차수는 더욱 감소된다는 것이 문제점으로 지적되고 있다.<sup>[19]</sup>

### 3. 이산화 문제와 수정한 설계기법

앞서 언급한 바와 같이 일반적으로 필터의 이상적 주파수응답은 주파수영역의 여러 대역에서 주어진다. 그러나 식 (5)의  $p$ 를 이 대역들에 포함된 무한한 점들의 집합이라 하면 식 (5)로 최대오차를 구하는 것은 불가능하고 또  $L.P.$  등을 이용  $p$ 상에서 최적의 이차원 연속계수 필터를 구하는 것도 불가능하다. 따라서 local search 기법이나 branch and bound 기법 모두 주파수영역을 적절히 이산화(discretization)하여 유한한 점들로 집합  $p$ 를 구성한 후  $p$ 상에서 최적 또는 준최적인 이차원 CSD계수 필터를 구하게 된다.<sup>[6]</sup> 그러나 이차원 주파수평면을 이산화하여 구한 최적 이차원 필터는 전체 주파수영역상의 최적 이차원 필터에 비해 바쁜 성능을 갖게 되고 이산화로 추출된 점의 수가 적을수록 그 오차가 증가한다.<sup>[19]</sup> 이는 이차원 CSD계수 필터에 있어서도 마찬가지이다. 그러므로 가능한 이차원 주파수평면에서 많은 점을 추출하여 이산화로 인한 성능저하를 막아야 한다. 이산화의 패턴으로는 여러 가지가 있으나 본 논문에서는 등간격 직교형 이산화를 가정한다. 필터의 차수가 증가할수록 그 주파수응답은 변화가 심하기 때문에 추출된 점간의 간격은 설계하고자 하는 이차원 필터의 차수에 따라 변화해야 한다.<sup>[19]</sup>  $(2M+1) \times (2N+1)$  이차원 필터설계의 경우  $u$  축 방향으로  $\pi/MG$ ,  $v$  축방향으로  $\pi/NG$ 의 간격으로 이산화할때  $G$ 가 약 10이상이면 이산화에 의한 오차를 무시할 수 있다.<sup>[19]</sup> 그러나  $(2M+1) \times (2N+1)$  이차원 필터의 경우 주파수영역의  $[0, \pi] \times [-\pi, \pi]$ 에서 약  $2NMG^2$  개의 점을 추출하나 같은 수의 독립된 필터계수를 갖는  $(4MN+2M+2N+1)$ 차 일차원 필터 설계의 경우 주파수영역  $[0, \pi]$ 에서  $G(2MN+M+N)$ 개의 점을 추출하므로  $G$ 가 10이상이면 이차원 필터설계시 집합  $p$ 의 원소 개수가 일차원 필터 설계의 경우보다 크게 증가한다.

이 같은 현상은 높은 차원의 필터를 설계함에 따라 가

증되고 집합  $p$ 의 원소갯수가 증가하면 필터설계의 소요 시간이 증가한다. 특히 L.P.나 ascent 알고리즘에서는 제약조건의 증가에 따라 수행시간이 급격히 증가하여 이차원 필터설계는 일차원 필터 설계에 비해 많은 시간을 요구한다. 이에 본 논문에서는 이차원 CSD계수 필터 설계시 L.P.나 ascent 알고리즘을 1회 이상 사용하는점에 착안 필터의 성능저하를 줄이며  $p$ 집합의 원소를 감소하여 설계 소요시간을 감소하는 방법을 제안한다.

제안한 방법의 설명을 위해 이차원 FIR 필터 설계의 문제를 다시쓰면 다음과 같다.

$$\begin{aligned}
 & \text{Find } h'(m,n)\text{'s that minimize} \\
 & \Delta = \sup_{(u,v) \in \rho} \{E(u,v)\} = \sup_{(u,v) \in \rho} \{W(u,v)D(u,v) - W(u,v)H(u,v)\} \\
 & \text{where} \\
 & E(u,v) = \left| \sum_{m=1}^I \sum_{n=1}^J h'(m,n) \cos(mu + nv) + \sum_{n=0}^J h'(0,n) \cos(nv) - D(u,v) \right| W(u,v). \quad (7)
 \end{aligned}$$

이를 행렬식으로 다음과 같이 다시 쓸 수 있다.

$$\begin{aligned}
 & A \in \mathbf{R}^{m \times n}, x \in \mathbf{R}^n, b \in \mathbf{R}^m, m \geq n. \\
 & \text{Define } r = \vec{b} - A \vec{x} \text{ where } \vec{r} \in \mathbf{R}^m. \\
 & \text{Find } x \text{ that minimizes } \Delta(\vec{x}) = \sup_{i \in \{1, \dots, m\}} \{|r_i|\} \quad (8) \\
 & \text{where } r = [r_1, r_2, r_3, \dots, r_m]^T.
 \end{aligned}$$

위에서  $m=n$  즉  $A$  행렬이 정방행렬이고  $\text{rank}(A)=m$  이면  $A$  행렬의 역행렬이 존재하고 따라서  $r$  벡터를 0으로 하는  $x$ 를  $x=Ab$ 식으로 구할 수 있다. 그러나  $m>n$ 이고  $\text{rank}(A)>n$ 이면  $r$  벡터를 0으로 하는 즉  $m$ 개의 식을 모두 성립하는  $x$ 는 없다. 그러나 각 식의 오차중 최대값이 최소가 되도록 하는  $x$ 는 존재한다. 이러한  $x$ 를 구하는 방법 중 필터 설계를 위해 많이 사용되는 것이 ascent 알고리즘이다.<sup>[20]</sup> 식 (7)과 (8)을 비교하면  $A$  행렬은  $W(u,v) \cos(mu + nv)$  값들로 이루어짐을 알 수 있다. 그러므로  $A$  행렬의 행의 수는 집합  $U$ 의 원소의 수와 같고 열의 수는 이차원 FIR 필터의 독립적인 필터계수의 수와 같다.

여기서 ascent 알고리즘의 근간이 되는 정리 1을 이용하면 실제로 최적 연속계수 이차원 필터를 설계하는데 필요한  $A$  행렬의 행의 수는 최대 독립적인 필터계수의 수보다 하나 많음을 알 수 있다.<sup>[20]</sup>

### Theorem 1.

*Every minimax solution of the system  $Ax = b$  is a minimax solution of an appropriate subsystem comprising  $N+1$  rows of  $A$ .*

즉 집합  $p$ 의 원소수에 관계없이 원소수가 독립적인 필터계수의 수보다 하나 많은 원소를 갖는 적절한 부분집합  $p'$ 를 취하면  $p'$ 상의 최적 연속계수 필터는  $p$ 상의 최적 연속계수 필터와 일치한다. 이러한 성질을 갖는 적절한 부분집합  $p'$ 를 구하는 것이 ascent 알고리즘이며 집합  $p'$ 상에서 필터 주파수응답의 최대오차가 발생한다.<sup>[6,7]</sup> 최적 연속계수 이차원 필터의 설계시에는 L.P.나 ascent 알고리즘의 1회 수행만으로 설계가 이루어지므로 이러한 성질을 이용 수행시간을 줄일 수 없다. 그러나 CSD계수 필터설계를 위한 local search 기법이나 branch and bound 기법에서는 반복적으로 집합  $p$ 를 사용하므로 집합  $p'$ 를 이용할 수 있다.

Branch and bound 기법에서는 최초로 이차원 연속계수 FIR 필터의 설계를 위해 L.P. 또는 ascent 알고리즘을 사용하고 이후로 고정되지 않은 필터 계수를 최적화하기 위해 L.P.나 ascent 알고리즘을 반복 사용한다. 한편 최적 연속계수 필터의 필터계수가 가까운 CSD로 고정될 때 발생하는 필터계수의 최적 연속계수 필터와 같이  $p'$ 상에서 발생한다. 이러한 경우에는  $p$ 상에서 구한 최적 CSD계수 필터와  $p'$ 상에서 구한 최적 CSD계수 필터와 일치한다. 따라서 연속 필터계수가 CSD로 고정되며 발생하는 변화가 충분히 미세하다고 가정하면 이차원 연속계수 필터를 설계하면서 구한  $p'$ 를 이용하여 이후에  $p'$ 상에서 고정되지 않은 필터계수의 최적값을 구함으로써  $p$ 나 ascent 알고리즘의 수행시간을 단축할 수 있다. 그러나 집합  $p'$ 는 이차원 주파수영역에서 밀집하지 않으므로 수형도의 각 절점에서 최대오차의 하한을 구할 때 실제의 하한보다 작은 값을 얻게 된다. 이는 tree 구조에서 불필요한 분기를 크게 증가시켜 비록 일회성의 L.P.나 ascent 알고리즘 수행시간을 단축되나 수행횟수가 크게 증가해 전체 소요시간은 단축되지 않는다.<sup>[19]</sup> 이러한 문제점은 각 절점에서 고정되지 않은 필터계수를  $p'$ 상에서 구하고 필터의 최대오차를  $p$ 상에서 구해 실제의 최대오차 하한 이상의 값을 절점에 할당함으로써 불필요한 분기를 감소해 해결할 수 있다. 한편 실제로 연속 필터계수를 CSD로 고정할 때 발생하는 변화는 일반적으로 충분히 미세하지 않아 앞서 언급한 방법으로 구한 CSD계수 필터

가 최적임을 보장할 수 없다. 이에 실험을 통해 수정된 branch and bound 기법의 성능과 설계 소요시간을 기존의 branch and bound 기법과 비교하였다.

Local search 기법에서는 최초로 이차원 연속계수 필터를 설계할 때를 제외하면 L.P.나 ascent 알고리즘을 사용하지 않으므로 집합 p를 임시해에 이웃하는 여러 CSD계수 필터의 성능을 비교하기 위한 기준 점들의 집합으로 이용하게 된다. 따라서 local search 기법에서 집합 p 대신 집합 p'상에서 임시해에 이웃하는 필터들의 성능을 비교할 경우 각 필터의 최대오차를 구하기 위해 오차를 계산해야 할 점의 수가 감소하게 된다. 이는 local search 기법의 요소시간을 단축하는 요인이 되며 비교되는 필터간의 성능 비교는 비교 기준점의 수가 감소한 만큼 부정확하게 되어 일반적으로 결과로 얻은 CSD계수 필터의 성능은 저하된다. 이에 성능 저하의 정도와 계산량 감축의 효과를 알아보기 위해 실험을 통해 기존의 local search 기법과 수정된 local search 기법을 비교하였다.

### III. 실험 결과 및 검토

#### 1. 적절한 이차원 대각 보간 필터

대각표본추입이란 이차원 이산신호(digital signal)의 그림 1과 같은 대각구조의 표본추입이다.<sup>[2,10]</sup> 이때 인접한 고주파대역의 스펙트럼과 겹침(aliasing)을 방지하기 위해 겹침방지(anti-aliasing) 필터링이 필요하다.<sup>[2]</sup> 일반적으로 영상신호는 그 대부분의 에너지를 저주파대역에 포함하고 있고 인간의 시각이 대각방향의 고주파성분보다 수평, 수직방향의 고주파성분에 민감하다는 점을 고려하여 일반적으로 마름모형태의 필터가 많이 사용되고 있다.<sup>[3]</sup> 따라서 본 논문에서는 겹침방지필터와 보간필터의 형태는 마름모형태라 전제한다.

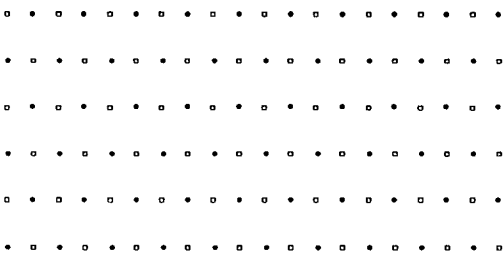


그림 1. 이차원 대각 표본추입  
Fig. 1. 2-D quincuncial subsampling.

실험을 통해 적합한 필터 특성을 조사하기 위해 그림 2에 마름모형태의 필터설계에 사용할 이상적 주파수응답의 가중치모델과 상수를 정의 도시하였다. 그림 2에서 (0,0), ( $\pm\pi, \pm\pi$ )부근에서의 가중치를 모두 1로 고정하였다. 영상신호의 에너지중 대부분이 (0,0) 부근의 저주파성분에 집중되어 있으므로 통과대역의 가중치  $W_p$ , 차단대역(stopband)의 가중치  $W_s$ 는 1보다 매우 작은 값이 되어야 한다. 또한  $\alpha, \beta$ 는 각각 원점으로부터 통과대역, 차단대역까지의 거리를 나타내는 변수이다. 이차원 필터의 최대오차와 변화영역(transition area)의 면적은 트레이드오프(trade off) 관계가 있다. 따라서 적절한 대각보간 필터의 특성을 찾기 위해 적절한 변화영역의 폭을 구해야 한다. 또한 통과대역과 차단대역의 가중치에 따라 각 대역의 최대오차도 변화하므로 이들 대역의 적절한 가중치도 구해야 한다. 일반적으로 필터의 차수가 증가하면 보간영상의 PSNR도 증가한다. 그러나 이상적인 필터를 사용해도 PSNR은 무한하게 증가하지 않는다. 이는 표본추입 및 보간시에 원 영상신호가 대역한정(band limited)되어 있지 않다면 반드시 손실되는 신호가 있기 때문이다.<sup>[15]</sup> 따라서 적절한 대각 보간 필터의 차수를 결정하기 위해 여러 차수의 이차원 필터를 설계하여 가장 적절한 차수를 결정하여야 한다. 이에 본 논문에서는  $3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9, 11 \times 11$ 의 다른 차수의 이차원 필터를 각각의 차수마다  $\alpha, \beta, W_p, W_s$ 를 여러가지 값으로 변화하여 설계하고 대각 보간시의 성능을 비교하였다. 필터는 ascent 알고리즘을 이용한 이차원 필터 설계기법으로 설계하였으며 주파수영역을  $63 \times 63$ 의 직교격자로 추출하여 필터의 제약조건을 부가하였다.<sup>[6,20]</sup> 대각 표본추입에는 2:1, 8:1등 여러종류가 있으나 가장 대표적인 2:1의 표본

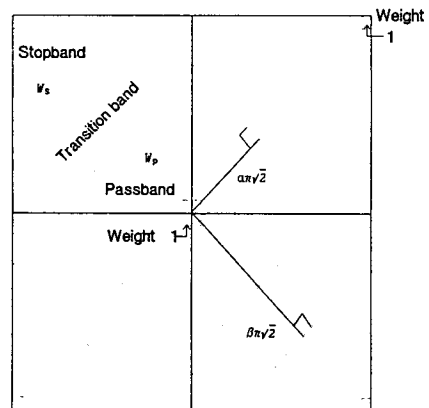


그림 2. 대각 보간필터 설계에 사용될 가중치 모델  
Fig. 2. Weight model for quincuncial interpolation filter design.

줄임비로 실험하였다. 참고로 2:1의 표본줄임은 HD-MAC방식에서는 움직임이 적은 블록의 전송을 위해 사용된다.<sup>[1]</sup> 단일 영상신호의 실험으로는 적절한 필터 특성을 찾을 수 없으므로 512×512의 원영상을 4:1로 256×256의 Lena, Boats, Sail, Bridge, Pepper의 5가지 영상신호를 사용 실험하였다. 또한 HD-MAC등 여러가지 응용분야에서 영상신호를 블록단위로 처리하는 점을 감안 보간시 8×8 블록단위로 보간하여 얻은 영상신호의 원영상과의 PSNR로 필터의 성능을 비교하였다.

표 1,2,3,4의 결과를 보면 3×3 필터의 경우 최고 30.1 dB, 5×5 필터의 경우 최고 31.26 dB, 7×7 필터의 경우 최고 31.66 dB, 9×9 필터의 경우 최고 31.76 dB의 성능을 각각 보였다. 즉 9×9 필터는 7×7 필터와 비교 평균 0.1 dB의 미세한 성능향상을 보였으며 이는 이 이상의 필터차수 증가로 큰 성능향상이 없음을 보여준다. 따라서 8×8 블록단위 2:1 보간필터의 차수는 7×7 정도가 적절함을 알 수 있다. 또한 표 3에서 최고의 성능을 7×7 필터를 찾으면 8×8 블록단위로 2:1 보간필터로 적합한 필터는  $\alpha=0.314$ ,  $\beta=0.686$ ,  $W_p=0.003$ ,  $W_s=0.01$ 로 설계된 7×7 이차원 FIR 필터임을 알 수 있다.

표 1. 3×3 필터의 성능

Table 1. Performance of 3×3 filter.

$W_p$ $W_s$	$\alpha$	0.196	0.235	0.275	0.314	0.353
	$\beta$	0.804	0.765	0.725	0.686	0.647
0.001,0.01		29.68	30.01	30.10	29.93	29.74
0.003,0.01		29.68	30.01	30.10	29.99	29.94
0.01, 0.01		29.17	29.20	29.17	29.19	29.03
0.01,0.003		28.34	28.09	27.49	26.17	23.95
0.01,0.001		27.75	27.18	26.43	25.04	25.74

표 2. 5×5 필터의 성능

Table 2. Performance of 5×5 filter.

$W_p$ $W_s$	$\alpha$	0.196	0.235	0.275	0.314	0.353
	$\beta$	0.804	0.765	0.725	0.686	0.647
0.001,0.01		31.04	31.25	31.26	31.05	30.74
0.003,0.01		30.57	30.93	31.24	31.23	30.98
0.01, 0.01		30.49	30.69	31.04	31.18	30.62
0.01,0.003		30.26	30.50	30.73	30.89	30.31
0.01,0.001		30.26	30.36	30.21	30.41	29.68

표 3. 7×7 필터의 성능

Table 3. Performance of 7×7 filter.

$W_p$ $W_s$	$\alpha$	0.196	0.235	0.275	0.314	0.353
	$\beta$	0.804	0.765	0.725	0.686	0.647
0.001,0.01		31.26	31.50	31.63	31.55	31.48
0.003,0.01		30.93	31.14	31.34	31.66	31.65
0.01, 0.01		30.82	30.96	31.07	31.44	31.38
0.01,0.003		30.75	30.85	30.94	31.14	31.12
0.01,0.001		30.68	30.76	30.82	31.09	31.18

표 4. 9×9 필터의 성능

Table 4. Performance of 9×9 filter.

$W_p$ $W_s$	$\alpha$	0.196	0.235	0.275	0.314	0.353
	$\beta$	0.804	0.765	0.725	0.686	0.647
0.001,0.01		31.67	30.74	31.17	31.76	31.73
0.003,0.01		31.67	30.37	31.00	31.71	31.72
0.01, 0.01		30.32	31.02	31.12	31.48	31.54
0.01,0.003		30.32	30.98	31.12	31.35	31.44
0.01,0.001		30.83	30.85	31.17	31.14	31.23

## 2. 각 설계 기법들의 비교 및 검토

앞서 언급했듯이 기존 설계기법의 계산량을 감축하기 위해 수정된 설계기법의 비교를 위해 실험을 통해 이들 설계기법의 성능과 소요시간을 비교하였다. 비교의 목적은 소요시간을 단축하기 위해 수정된 기법을 이용 이차원 CSD계수 FIR 필터를 설계했을때 설계 소요시간의 단축 정도와 설계된 필터의 성능을 비교하는데 있다. 이와 같은 비교는 local search 기법과 branch and bound 기법 모두에 대해 필요하며 설계된 필터의 성능과 설계 소요시간은 설계하고자 하는 필터에 따라 매우 불규칙하게 변화하므로 여러 종류의 필터에 대해 실험하여야 한다. 따라서 본 논문에서는 비교적 짧은 시간에 구할 수 있는 3×3, 5×5의 두가지 차수의 필터를 각각 3가지 다른 이상적 주파수응답과 가중치로 설계하여 얻은 필터의 최대오차와 소요시간을 비교하였다. 이 실험에서 사용한 필터의 3가지 이상적 주파수 응답과 가중치는 그림 3에 도시하였으며 실험 결과를 표 5에 제시하였다. Branch and bound 기법에서는 최적연속계수 필터의 각 계수를 그집합 5개의 CSD로 대체하여 분기하였으며 수정된 기법에 있어서도 같은 방법을 사용했다. 또한 분기할 결점을 결정할 방법으로는 depth-first 기준을 단독으로 사용했다.<sup>[7,19]</sup>

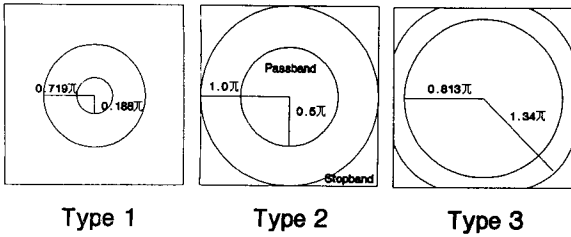


그림 3. 무곱셈 필터 설계에 사용된 가중치 모델  
Fig. 3. Weight model for multiplierless filter design.

Local search 기법에서는 한번에 필터 계수중 두 계수만을 변화하여야 탐색했다. 또한 local search 기법 및 branch and bound 기법의 실험에서 공통적으로  $L=2$ ,  $\ell=12$ 로 정의된 CSD를 사용하였고 scaling factor의 최적화는 시도하지 않았으며 단순히 연속계수 필터의 필터계수와 CSD간의 제곱오차합을 최소화하scaling factor를 사용했다. 직교형 격자로 이차원 주파수 영역을 이산화하였으며 필터의 차수에 관계없이  $[-\pi, +\pi] \times [-\pi, +\pi]$ 에서  $63 \times 63$ 개의 점을 추출하였다.

우선 표 5의 결과를 통해 설계된 필터의 성능을 비교하면  $3 \times 3$  필터의 경우 기존의 branch and bound 기법과 수정된 branch and bound 기법은 3가지 필터 모두에서 같은 결과를 나타냈다. 또한 기존의 local search 기법도 같은 결과를 나타냈으나 type 1, type 3의 필터에서 branch and bound 기법보다는 저하된 성능을 보였다.  $5 \times 5$  필터의 경우 설계 기법간의 성능차가  $3 \times 3$  필터의 우보다 분명히 나타났으며 type 3의 경우를 제외하면 기존의 branch and bound 기법과 수정된 branch and bound 기법의 성능차는 0.5 dB 이내로 미세했다. 기존의 local search 기법과 수정된 branch and bound 기법의 성능차는  $3 \times 3$  필터설계시보다 증가되어 평균 1.6 dB 이상의 격차가 있었다. 한편 기존의 local search 기법과 수정된 local search 기법의 비교에서는 branch and bound 기법에서와 마찬가지로 0.5 dB 이내의 미세한 격차가 있었다.

소요시간면에서 각 설계 기법을 비교해 보면 기존의 branch and bound 기법, 수정된 branch and bound 기법, 기존의 local search 기법, local search 기법순으로 소요시간이 많았다. 또한 필터 차수의 증가에 따른 소요시간의 증가속도도 위와 같은 순서로 감소되었으며 특히 수정된 local search 기법의 경우 6가지 필터 모두에 대해 30초 미만의 시간을 소요해 가장 작은 설계 소요시간을 보였다. 3 종류의  $5 \times 5$  필터의 설계 소요시간을 평균하면 기존의 branch and bound 기법이 약 2400초, 수정

표 5. 무곱셈 필터 설계기법의 성능 비교  
Table 5. Performance comparison of design techniques.

		branch and bound	modified branch and bound	local search	modified local search
Type1 3×3	stopband ripple	-16.70 dB	-16.70 dB	-16.66 dB	-16.66 dB
	passband ripple	-17.39 dB	-17.39 dB	-17.42 dB	-17.42 dB
	consumed time	17 sec	16 sec	10 sec	3 sec
Type2 3×3	stopband ripple	-9.34 dB	-9.34 dB	-9.34 dB	-9.34 dB
	passband ripple	-9.39 dB	-9.39 dB	-9.39 dB	-9.39 dB
	consumed time	4 sec	3 sec	6 sec	2 sec
Type3 3×3	stopband ripple	-12.44 dB	-12.44 dB	-12.50 dB	-12.50 dB
	passband ripple	-12.42 dB	-12.42 dB	-12.80 dB	-12.80 dB
	consumed time	14 sec	13 sec	7 sec	1 sec
Type1 5×5	stopband ripple	-18.86 dB	-18.86 dB	-18.60 dB	-18.11 dB
	passband ripple	-18.85 dB	-18.84 dB	-18.69 dB	-18.69 dB
	consumed time	2547 sec	406 sec	86 sec	14 sec
Type2 5×5	stopband ripple	-22.37 dB	-22.52 dB	-21.22 dB	-21.20 dB
	passband ripple	-22.46 dB	-22.03 dB	-21.41 dB	-21.26 dB
	consumed time	4297 sec	819 sec	38 sec	5 sec
Type3 5×5	stopband ripple	-27.34 dB	-25.92 dB	-21.85 dB	-21.85 dB
	passband ripple	-27.18 dB	-25.83 dB	-21.63 dB	-21.63 dB
	consumed time	375 sec	323 sec	50 sec	6 sec

된 branch and bound 기법이 약 510초, 기존의 local search 기법이 43초, 수정된 local search 기법이 약 8초였다. 설계에 사용된 컴퓨터는 MIPS사의 RS 2030

workstation이며 모든 실험은 단일 부하상태에서 이루어졌다. 즉 local search 기법이나 branch and bound 기법 모두에서 수정된 기법이 약 1/5의 소요시간 단축이 있었으나 3×3 필터설계시의 소요시간 단축과 비교하면 필터의 차수 증가에 따라 단축비가 더욱 감소할 것을 알 수 있다.

이상과 같은 결과로부터 제안된 수정기법은 branch and bound 기법과 local search 기법과 비교할때 성능은 미세하게 저하되나 소요시간을 크게 단축할 수 있음을 알 수 있다. 한편 설계할 수 있는 필터의 최대 차수는 기존의 branch and bound 기법, 수정된 branch and bound 기법, 기존의 local search 기법, 수정된 local search 기법 순으로 증가하고 설계된 필터의 성능은 이의 역순으로 우수하다.

3. 무곱셈 필터의 성능 평가

CSD계수 필터의 성능이 같은 차수의 연속계수 필터의 성능에 못 미치는 것은 분명하나 그 성능저하의 정도를 알아보고 대각보간시 CSD계수로 설계된 무곱셈 이차원 필터의 성능을 연속계수 이차원 필터의 성능과 비교하기 위해 다음과 같은 실험을 하였다. 앞장에서 구한 대각보간에 적합한 가중치를 사용하여 설계한 7×7 연속계수 필터와 수정된 branch and bound 기법으로 설계한 7×7 CSD계수 필터를 통과대역, 차단대역의 오차로 비교하여 그 결과를 표 6에 제시하였다. 또 Boats, Sail, Lena, Bridge, Pepper 영상신호를 사용하여 앞서 비교한 7×7 연속계수 필터, 7×7 CSD계수 필터, 그리고 객관적인 비교기준으로써 일본 NHK의 MUSE 방식에서 사용하는 7×5연속계수필터로 보간한 영상신호의 PSNR도 함께 비교하여 그 결과를 표 7에 제시하였다. 이때 CSD계수 필터는 앞절과 마찬가지로  $l=12, L=2$ 로 하여 설계되었다. 표 6,7의 결과를 보면 7×7 CSD계수 필터의 통과대역 오차가 약 2.12 dB, 차단대역 오차가 약 7.55 dB정도 연속계수 필터에 못미치는 것을 알 수 있으나 실제로 영상신호의 보간에 사용하여 얻은 결과는 큰 차이를 보이지 않음을 알 수 있다. 또한 7×5 연속계수 필터로서 좋은 성능을 인정받고 있는 MUSE 방식의 필터보다 7×7 CSD계수 필터가 향상된 성능을 보임을 알 수 있다. 이는

표 6. 7×7 연속계수 필터와 무곱셈 필터의 비교  
Table 6. Comparison of continuous coefficient filter and multiplierless filter of order 7×7.

	7×7 연속계수 필터	7×7 무곱셈 필터
통과대역 오차	-25.97 dB	-23.85 dB
차단대역 오차	-36.64 dB	-29.09 dB

표 7. 여러가지 대각보간 필터의 성능  
Table 7. Performance of interpolation filters.

	Boats	Sail	Bridge	Lena	Pepper	Aver.
7×7 연속계수	31.90 dB	30.52 dB	29.09 dB	33.05 dB	33.76 dB	31.66 dB
7×7 무곱셈	31.60 dB	30.29 dB	28.87 dB	32.86 dB	33.56 dB	31.44 dB
7×5 연속계수	31.71 dB	30.09 dB	28.50 dB	33.02 dB	33.30 dB	31.32 dB

CSD계수라는 제약에서 오는 성능저하를 필터의 차수를 증가시킴으로써 충분히 제거하고 오히려 성능을 향상시킬 수 있다는 것을 의미한다. 또한 비록 필터의 차수가 증가한다 해도 필터계수가 CSD이므로 곱셈이 필요없어 설계 구현은 용이해진다. 예를 들면 7×5 연속계수 필터의 구현시 한 화소당 12번의 곱셈과 34번의 덧셈이 필요하나 7×7 CSD계수 필터의 경우 한 화소당 최대 64번의 덧셈만이 필요해 구현이 용이함을 알 수 있다. 참고로 그림 4에 MUSE에서 사용하는 7×5 연속계수 이차원 필터와 수정된 branch and bound 기법으로 설계한 7×7 CSD계수 이차원 필터의 주파수 응답을 도시하였다.

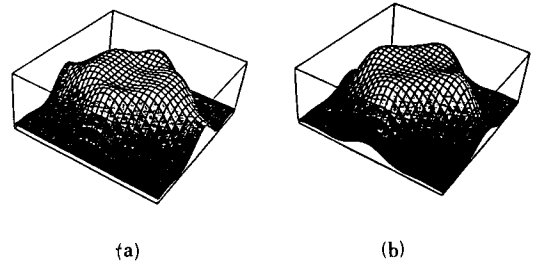


그림 4. 이차원 대각 보간 필터의 주파수 응답  
(a) 7×5 연속계수 필터  
(b) 7×7 무곱셈 필터

Fig. 4. Frequency response of interpolation filter.  
(a) 7×5 continuous coefficient filter,  
(b) 7×7 multiplierless filter.

IV. 이차원 무곱셈 필터의 VLSI 구현

영상신호처리분야에서 이차원 FIR 필터의 응용은 매우 중요한 위치를 차지하고 있다. 이에 지난 수년간 이차원 필터의 실시간 구현기법에 관한 연구가 소개되고 있다. [4,11,12,13,14] 이들 연구 결과를 간단히 살펴보면 대부분 실시간구현에 필요한 하드웨어(hardware)의 감축과 처리속도의 향상을 목적으로 하며 이를 위해 파이프라인 구조와 병렬처리를 주로 활용하고 있다.



AP는 구현 구조의 성능과 복잡도를 비교하기 위해 널리 쓰이는 비교 기준이다. 여기서 A는 VLSI 구현시 차지하게되는 면적을 의미하고 P는 처리주기 즉 입력되는 데이터의 주기를 의미한다. 최근에 발표된 Wu의 이차원 연속계수 필터 실시간구현 구조는  $O((2N+1)(2M+1)W^2)$ 의 AP를 가진다.<sup>[4]</sup> 기존의 다른 기법의 AP를 보면 대부분  $O((2N+1)(2M+1)W^3)$  이상의 수준으로 Wu의 구조가 w 정도의 성능 향상을 얻을 수 있다.<sup>[4]</sup> 여기서  $(2N+1)(2M+1)$ 은 이차원 필터의 차수이며 w는 데이터의 비트수이다.

효율적인 구현 구조와 VLSI 구현 적합성등 연속계수 이차원 필터 구현시 요구되는 조건은 무곱셈 즉 CSD계수 필터의 실시간 구현에서도 요구된다.

Wu의 구조는 블록단위의 병렬처리를 전제로하고 있다.

<sup>[4]</sup> 블록단위 병렬처리란 그림 5과 같이 영상신호를 여러 블록으로 분리하여 하나 또는 그 이상의 영상신호처리부에서 처리하는 방식으로 영상신호처리부를 증가시켜 처리속도를 증가시킬 수 있다는 장점이 있다. 그러나 중앙제어부가 필요하다는 점, 여러가지 제어신호와 동기신호가 존재한다는 점, 영상신호의 보관을 위해 많은 양의 기억소자를 필요로 하는 점등의 단점이 있다. 따라서 전체 영상신호에 동일한 영상신호처리를 할때는 가능하면 그림 6과 같이 전체 영상신호가 입력됨과 동시에 하나의 영상신호처리부에서 처리하는 것이 유리하다. 이에 본 논문에서는 좋은 성능을 인정받고 있는 Wu의 구조를 무곱셈 이차원 필터에 적합하도록 수정하여 블록단위 병렬처리 구조를 제안하고 또한 라인별로 입력되는 전체 영상신호의 순차적 처리를 위한 이차원 무곱셈 FIR 필터 구조를 제안한다.

1. 블록단위 병렬처리 구조

Wu가 제안한 systolic array의 단위(unit) 셀(cell)은 그림 7 a)와 같이 화소값과 필터계수의 곱을 구해 누산하는 기능을 한다.<sup>[4]</sup> 단위 셀의 입력은 shift register를 통해 비트직렬워드병렬로 변환된 화소값이며 화소값과 필터계수의 곱을 누산한 출력도 LSB(least significant bit)부터 비트직렬로 출력된다. 따라서 Wu의 단위 셀은 그림 7 b)와 같이 w 비트 덧셈기와 register로 구성되며 직렬로 입력되는 화소값에 따라 필터계수를 이동하여 더하여 LSB로부터 한 비트씩 순차적으로 출력하게 된다.

무곱셈 필터계수 즉 CSD를 다른 수와 곱할 때는 CSD를 이동시켜 더함으로써 곱셈을 구현하는 것이적은 수의 덧셈을 요구하므로 유리하다. 따라서 무곱셈 필터의 한계수 즉 CSD가 식 (6)과 같이 표현될 때 입력 화소값과 필터계수의 곱을 누산할 단위 셀을 그림 7 c)와 같이 수

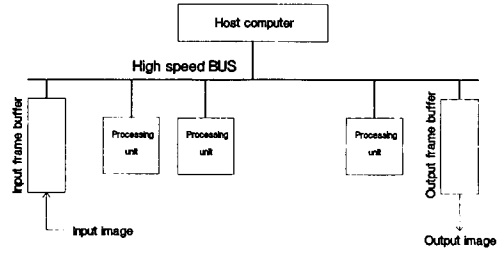


그림 5. 블록단위 병렬처리  
Fig. 5. Block parallel processing.

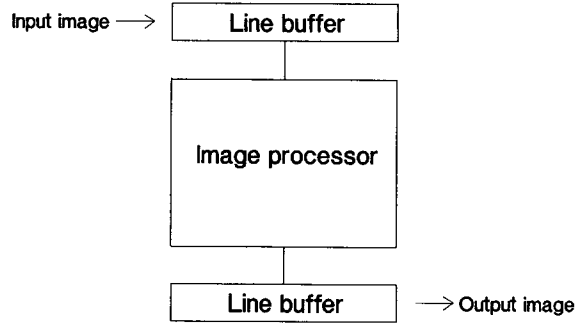


그림 6. 라인단위 병렬처리  
Fig. 6. Scan-line parallel processing.

정하므로써 무곱셈 필터의 구현상 잇점을 활용할 수 있다. 한편 이와같은 수정으로 단위 셀 이외의 구조는 Wu의 구조를 수정하지 않고 사용할 수 있다. 따라서 단위 셀만을 비교하여 위와 같은 수정으로 얻은 무곱셈 이차원 FIR 필터의 블록단위 병렬처리 구조와 연속계수 필터의 블록단위 병렬처리 구조를 비교할 수 있다. 그림 7의 단위 셀 구조를 살펴보면 연속계수 필터의 경우 한개의 w 비트 덧셈기, w 비트 register, 1 비트 덧셈기 또는 뺄셈기가 필요함을 알 수 있다. 무곱셈 필터계수의 경우 단위 셀은 총 L개의 덧셈기 또는 뺄셈기와 w+L개의 래치로 이루어진다. 덧셈기와 뺄셈기는 같은 정도의 복잡도를 갖고 있다고 가정하고 w비트 덧셈기는 w개의 1비트 덧셈기와 동일하다 할때 무곱셈 필터의 단위 셀이  $L/(w+1)$  정도 적은 수의 덧셈기를 갖고 있으며 요구되는 기억소자의 수도 증가하지 않음을 알 수 있다. 이와 같은 이차원 무곱셈 FIR 필터의 구조를 연속계수의 경우와 AP로 비교하면 약  $L/w$  수준의 감소를 기대할 수 있다. 이는 앞서 언급한 이론적 곱셈기에서의 하드웨어 감소율  $(L-1)/w$ 에 근접한 것으로 전체적인 AP는  $O((2N+1)(2M+1)Lw)$ 이 된다. 예를 들면  $w=20, L=2$ 인 경우를 고려하면 연속계수 필터에 비해 약 1/10의 하드웨어 감축이 가능하다.

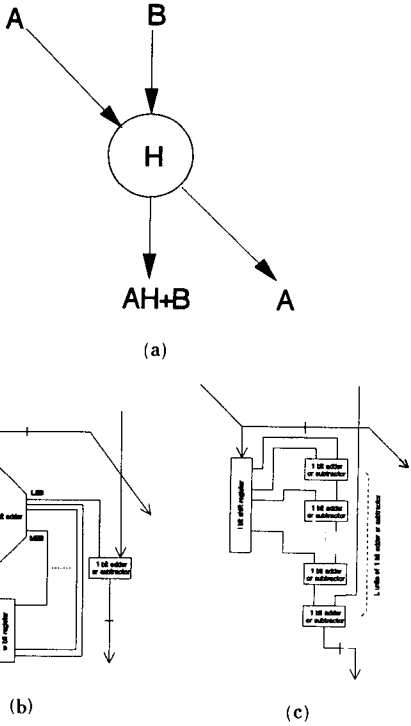


그림 7. 단위 셀의 구조  
 (a) 동작 원리  
 (b) 연속계수 필터의 경우  
 (c) CSD계수 필터의 경우

Fig. 7. Structure of PE.  
 (a) Basic operation,  
 (b) Continuous coefficient case,  
 (c) CSD coefficient case.

2. 라인단위 병렬처리 구조

이차원 FIR 필터의 라인단위 병렬처리 구조는 앞에서 사용한 단위 셀을 그림 8과 같이 배열함으로써 구현할 수 있다. 여기서도 블록단위 병렬처리시와 마찬가지로 비트 병렬워드지령으로 입력된 라인 입력을 비트지령으로 변환해야 한다. 따라서 shift register로 이루어진 버퍼가 입력을 비트지령형태로 변환해야하며 변환된 입력은 단위 셀로 이루어진 -M부터 M까지 2M+1개의 연산부로 각각 입력된다. m번째 연산부는 컨벌루션에서 필터계수의 한 행에 해당하는 연산을 담당하며 이를  $x(i,j)$ 를 입력 화소값  $y_m(i,j)$ 를 m번째 연산부에서 계산하여 누산할 화소값이라 할때 수식으로 표현하면 다음과 같다.

$$y_m(i,j) = \sum_{n=-N}^N h(m,n) x(i-m,j-n) \quad (9)$$

따라서 2M+1개의 연산부를 모두 통과하여 얻는 출력

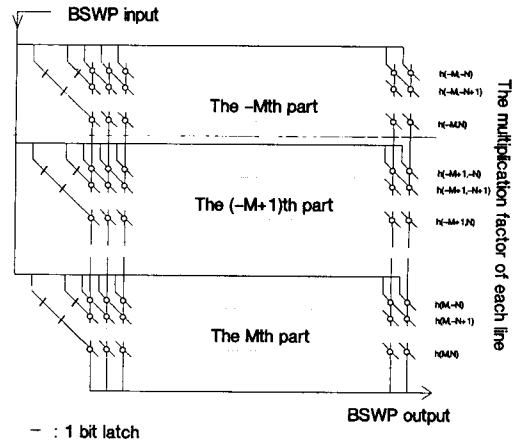


그림 8. 제안하는 라인단위 병렬처리 구조  
 Fig. 8. Structure for scan-line parallel processing.

즉  $y_m, \dots, y_m$ 의 합은 출력화소값이 된다. 이때 필요한 출력단위 버퍼는 비트지령 형태의 출력을 비트병렬 형태로 변환하여 출력하기 위한 것으로 입력버퍼와 반대의 역할을 한다.

이와 같은 라인단위 병렬처리 구조에 필요한 전체 단위 셀의 수는 입력 영상신호의 한 라인이 K개의 화소로 이루어졌다고 가정하면  $K(2N+1)(2M+1)^2$ 이 된다. 따라서 1 비트 덧셈기가 차지하는 면적을  $O(1)$ 라 하면 단위 셀이 차지하는 면적은  $O(L)$ 이 되고 전체 구조가 차지하는 면적은  $O(LK(2N+1)(2M+1))$ 이 된다. 한편 내부 파이프라인 주기를  $O(1)$ 이라 할때 입력 주기는  $O(w/K)$ 가 된다. 따라서 AP는  $O(wL(2N+1)(2M+1))$ 로 블록단위 병렬처리 구조와 같은 수준이며 연속계수 필터의 구현시보다  $1/w$ 의 간소가 있음을 알 수 있다. 한편 블록단위 병렬처리 구조에서와 마찬가지로 systolic array 구조이며 비트지령로 입력되는 화소값을 제외하고는 인접한 단위 셀과 연결이 이루어지므로 VLSI 구현에 용이하다.

라인단위 병렬처리로 이차원 필터를 구현할때 AP면에서는 블록단위 병렬처리시와 같다. 그러나 요구되는 하드웨어가 블록단위 병렬처리시보다 크게 증가한다. 예를 들어 라인당 256 화소를 갖는 입력 영상신호에  $7 \times 7$ 의 이차원 FIR 필터링을 라인단위 병렬처리로 구현한다 가정하면 필요한 전체 단위 셀의 수는  $256 \times 7 \times 7$ 이 된다. 연속계수 필터의 경우 단위 셀은 w 비트 덧셈기와 register로 이루어지므로 전체  $w \times 256 \times 7 \times 7$ 개의 1 비트 덧셈기와 같은 수의 1 비트 register 즉 래치가 필요하다. 여기서 w를 20이라 하면 약 25만 이상의 1 비트 덧셈기와 register가 필요하게 되며 이는 현재의 IC 집적기술로는 어려운 수준이다. 반면 앞서 언급했듯이 무공샘 필터의

경우 단위 셀이  $L$ 개의 1비트 덧셈기와  $w$ 개의 1 비트 register로 구성되므로 전체  $L \times 256 \times 7 \times 7$ 개의 1비트 덧셈기와  $w \times 256 \times 7 \times 7$ 개의 1 비트 register가 필요하다. 1비트 register의 경우 적은 수의 transistor로 구성되므로 구현이 어렵지 않고 일반적으로  $L$ 값은 2,3 정도에서 결정되므로  $L=2$ 라하면 전체적으로 필요한 1비트 덧셈기의 수가 2만5천개 수준이므로 현재의 IC 집적기술로 구현이 가능하다.

## V. 결 론

본 논문에서는 계산량을 단축한 무곱셈 필터 설계기법을 제안하였으며 또한 실제 대각 보간에 적합한  $7 \times 7$  이차원 무곱셈 필터를 설계하여 실제 영상신호에 적용하였다. 그 결과 연속계수 필터와 거의 성능차가 없었으며 MUSE 방식에서 사용되는  $7 \times 5$  연속계수 필터보다 우수한 성능을 보였다. 즉 필터계수가 CSD라는 제약으로 부터 발생하는 필터의 성능저하는 미세하며 필터의 차수를 증가함으로써 상대적으로 작은 차수의 연속계수 필터보다 성능을 향상할 수 있음을 알 수 있다. 따라서 이차원 필터링이 필요한 모든 분야에서 증가된 차수의 무곱셈 필터를 사용함으로써 구현 비용의 절감과 성능향상을 기대할 수 있다. 특히 구현 비용 증가로 인한 가격상승을 지양해야 하는 가전분야, 예를 들면 HDTV, video-phone 등의 분야에서는 무곱셈 필터의 사용이 성능 및 비용면에서 유리하며 특히 대각 보간을 사용하는 HD-MAC이나 MUSE 방식의 수상기에는 이차원 무곱셈 대각 보간 필터의 사용이 바람직하다.

## 參 考 文 獻

[1] F.W. Vreeswijk, "An HD-MAC coding system," Signal processing of HDTV, Elsevier silence publishing company INC., 1988.  
 [2] D.E. Dudgeon and R.M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, 1984,  
 [3] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.  
 [4] C.W. Wu, "Bit-level pipelined 2-D digital filters for real time image processing," *IEEE Trans. Circuit. Syst. for Video Tech.*, vol. 1, pp. 22-34, Mar. 1991.  
 [5] H. Samuelli, "An improved search algorithm

for the design of multiplierless FIR filters with powers of two coefficients," *IEEE Trans. Circuit. Syst.*, vol. CAS-34, pp. 1044-1047, July 1989.

- [6] Y. Kamp and J.P. Thiran, "Chevyshev approximation for two-dimensional nonrecursive digital filters," *IEEE Trans. Circuit. Syst.*, vol. CAS-22, pp. 208-218, Mar. 1975.  
 [7] Y.C. Lim and S.R. Parker, "FIR filter design over a discrete powers of two coefficient space," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 583-591, June 1983.  
 [8] Q. Zhao and Y. Tadokoro, "A simple design of FIR filters with powers of two coefficients," *IEEE Trans. Circuit. Syst.*, vol. CAS-35, pp. 566-570, May 1988.  
 [9] d. Kodek and K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital Filters," *IEEE Trans. Circuit. Syst.*, vol CAS-28, pp. 28-32, Jan. 1981.  
 [10] A. Lehtonen et al., "Nonlinear quincix interpolation filtering," *Visual Communications and Image Processing*, vol. 1360, pp. 135-142, Oct. 1990.  
 [11] K. Rank and R. Unbehauen, "Implementation of 2-D filters for real time processing," *IEEE Trans. Circuit. Syst.*, vol CAS-36, pp. 1252-1255, Sep. 1989.  
 [12] T. Aboulnasr and W. Steenaert, "Real time systolic array processor for 2-D spatial filtering," *IEEE Trans. Circuit. Syst.*, vol. CAS-35, pp. 948-957, Oct. 1986.  
 [13] K.M. Ty and a.N. Venetsanopoulos, "A fast filter for real time image processing," *IEEE Trans. Circuit. Syst.*, vol. CAS-33, pp. 948-957, Oct. 1986.  
 [14] A. Peled and B. Liu, "A new hardware realization of digital filters," *Trans. Acoust., Speech, Signal Processing*, vol. 22, pp. 456-462, Dec. 1974.  
 [15] A.V. Oppenheim and R.W. Schaffer, *Discrete Time Signal Processing*, Prentice-Hall, 1989.  
 [16] Y.C. Lim and S.R. Parker, "Discrete coefficient FIR digital filter design based upon an LMS criteria," *IEEE Trans. Circuit. Syst.*, vol. CAS-30, pp. 723-729, Oct. 1983.

著 者 紹 介



崔 鎮 宇(正會員)

1967年 12月 25日生. 1990年 2月 서울대학교 제어계측공학과졸업. 1992年 2月 서울대학교 대학원 제어계측공학과 졸업. 공학석사학위 취득. 현재 군복무중.

李 商 郁(正會員)

第 25 卷 第 7 號 參照  
현재 서울대학교 제어계측공학과 부교수