

논문 2008-45SP-2-8

# 실시간 시공 노이즈 제거 시스템 구현

## (Implementation of a Real-Time Spatio-Temporal Noise Reduction System)

홍혜정\*, 김현진\*, 강성호\*\*

(Hyejeong Hong, HyunJin Kim, and Sungho Kang)

### 요약

시공필터는 공간필터로는 제거할 수 없는 동영상의 노이즈를 제거하지만 알고리즘이 매우 복잡하여 하드웨어로 구현하기에 부적절하다. 본 논문에서는 적응평균필터 알고리즘을 바탕으로 최대 세 장의 프레임을 사용하는 실시간 시공 노이즈 제거 시스템을 구현한다. 기존의 알고리즘에서 하드웨어로 구현하기에 부적절한 요소들을 수정하였다. 동작 속도를 높이기 위해서 노이즈 추정과 필터링이 병렬적으로 수행되도록 이전 프레임에서 추정된 노이즈를 현재 프레임 필터링에 이용하게 하였다. 또한 필터링 윈도우의 형태를 변형하여 시스템의 동기화를 용이하게 하였다. 제안하는 구조는 Virtex 4 XC4VLX60 상에 구현하였고 총 66%의 슬라이스를 사용하고 최대 80MHz의 속도로 동작하였다.

### Abstract

Spatio-temporal filters are capable of reducing noise from moving pictures, which cannot be dealt with by spatial filters. However, the algorithm is too complicated to be realized as hardware. We implemented a real-time spatio-temporal noise reduction system using at most three frames, based upon adaptive mean filter algorithm. Some factors which interfere with hardware implementation were modified. Noise estimated from the previous frame was used to filter the current frame so that filtering could be conducted in parallel with noise estimation. This speeds up the system thereby enabling real-time execution. The form of filtering windows was also modified to facilitate synchronization. The proposed structure was implemented on Virtex 4 XC4VLX60, occupying 66% of total slices with 80MHz of the maximum operation frequency.

**Keywords:** 시공 노이즈 제거, 실시간, 적응평균필터, FPGA

## I. 서론

영상은 생성되는 순간부터 오염되기 시작한다. 카메라에 찍히면서, 저장매체에 저장되면서, 전송 채널을 통과하면서, 영상에는 노이즈가 더해진다. 노이즈는 영상의 질을 떨어뜨릴 뿐만 아니라 이후의 다른 처리를 어렵게 만든다<sup>[1]</sup>. 따라서 노이즈를 제거하는 전처리 과정은 영상처리에서 필수적이다.

일반적으로 사용되는 필터는 대개 공간필터이다. 공간필터는 대상이 되는 프레임 자체의 정보만으로 노이즈를 제거한다<sup>[2]</sup>. 모든 픽셀에 대해서 일정한 개수의 이

웃 픽셀들로 윈도우를 만들고, 내부의 픽셀들을 연산한 결과로 기존의 픽셀을 대체한다. 예를 들어, 대표적인 공간필터인 평균필터는 윈도우 안의 픽셀들의 평균으로 노이즈를 제거하려는 픽셀을 대체한다.

공간필터는 알고리즘이 단순하여 적은 비용으로 하드웨어를 구현할 수 있다는 장점이 있지만, 성능이 제한적이다. 사람의 눈은 시간적인 변화에도 민감하기 때문에<sup>[3]</sup> 공간필터를 거친 연속된 프레임을 동영상으로 재생하면, 개별적인 프레임에서는 보이지 않았던 노이즈가 새롭게 보이게 된다.

최근 뛰어난 성능의 노이즈 제거 알고리즘들이 제안되어 왔지만, 성능이 발전하는 만큼 점점 더 복잡해지고 있다. 성능이 개선되는 것은 이 알고리즘들이 고려하는 사항들이 많아졌기 때문이다. 더 많은 상황을 고

\* 정회원, \*\* 평생회원, 연세대학교 전기전자공학과  
(School of Electrical & Electronic Engineering,  
Yonsei University)  
접수일자: 2007년7월3일, 수정완료일: 2008년2월27일

려해서, 더 여러 번의 결정을 내려야 하므로 연산은 복잡해지고 연산의 양도 많아진다<sup>4~5)</sup>. 이들을 실제로 구현하려면 비용이 많이 들 뿐만 아니라 실시간으로 동작하기에 충분한 속도가 보장될지 의문이다.

게다가 개선된 성능을 실제로 영상을 보는 사람이 충분히 지각할 수 있는지도 문제가 된다. 사람의 시각적 지각 능력에는 한계가 있어서, 일정 수준 이상의 성능 차이는 인지되지 않는다. 영상의 질을 객관적으로 나타내기 위해서 PSNR(peak signal-to-noise ratio)과 같은 수치를 사용하지만, 결국 영상의 질은 보는 사람의 눈으로 판단한다<sup>12)</sup>. 복잡한 알고리즘으로 설계한 새 필터의 성능이 기존의 필터의 성능과 육안으로 구별하기 어렵다면, 새 필터 설계를 설계하는 데에 든 비용과 노력은 무의미하다.

본 논문에서는 알고리즘 자체의 뛰어난 성능보다는 하드웨어로 구현했을 때의 효율성과 실용성에 초점을 맞추어 시공 노이즈 제거 시스템을 구현하였다. 가우시안 노이즈를 제거하는 선형필터로 가장 대중적인 필터 중의 하나인 적응평균필터를 구현하되, 구현에 유리하도록 알고리즘을 수정하였다. 제안하는 구조는 HDL로 설계되어 FPGA(field programmable gate arrays) 위에 매핑되었다.

## II. 알고리즘

가우시안 노이즈를 제거하는 선형 필터 중에서 가장 대표적인 것이 평균 필터이다. 노이즈는 이웃 픽셀들과 비교하여 크게 다른 계조치(gray level)를 가진다. 이를 제거하기 위해서 평균필터는 픽셀을 중심으로 사각형 형태의 윈도우를 구성하고, 윈도우 내의 픽셀들의 평균으로 그 픽셀을 대체한다<sup>12)</sup>.

평균필터의 문제점은 영상 내의 경계를 흐리게 한다는 것이다. 영상에서 경계는 계조치가 다른 두 면이 맞닿는 부분인데, 평균필터를 지나면서 이웃 픽셀들의 영향에 의해 무더지게 된다. 이와 같이 영상이 흐려지는 효과를 막기 위해서 평균필터에 적응성을 도입했다. 적응평균필터는 윈도우 내의 모든 픽셀들로 평균을 구하지 않고, 일정 기준에 해당하는 픽셀만으로 평균값을 구한다. 즉, 경계는 필터링의 대상이 되지 않는다<sup>6)</sup>.

본 논문에서 구현하는 시공 노이즈 제거 시스템은 적응평균필터 알고리즘을 기반으로 하고, 세 장의 연속된 프레임을 사용하여 시공 필터링을 한다. 적응평균필터링 알고리즘은 두 단계로 이루어진다. 먼저 프레임에

노이즈가 얼마나 포함되어 있는지 추정한다. 추정된 노이즈는 다음 단계인 필터링 단계에서 윈도우 내의 각 픽셀들을 필터링에 사용할 것인지 아닌지 판단할 때 기준으로 사용된다.

### 1. 노이즈 추정 단계

적용필터는 상황에 따라 필터링의 대상, 범위 등을 조절하는데, 이에 필요한 결정을 내리기 위한 기준으로 사용되는 것이 현재 프레임에 포함된 노이즈의 양이다. 따라서 실질적인 필터링 과정 이전에 매 프레임에 대한 노이즈를 추정하는 과정이 필요하다.

먼저, 프레임을 작은 단위윈도우들로 나눈다. 윈도우 내에 경계가 없다면 픽셀들 간에 편차가 거의 없을 것이고, 경계가 있다면 편차가 클 것이다. 만약 작은 크기의 편차가 있다면, 이는 노이즈에 의한 것이다. 그러므로 프레임 내의 모든 윈도우의 표준편차 중에서 최소값을 찾아, 그 값을 현재 프레임의 노이즈로 결정한다.

### 2. 필터링 단계

이 적응평균필터는 윈도우 내의 픽셀 중에서 현재 픽셀과의 절대차가 문턱값보다 작거나 같은 픽셀만을 선택하여 현재 픽셀의 새로운 값을 계산하는 데에 사용한다. 여기서 문턱값은 이전의 노이즈 추정의 결과에 3을 곱한 값을 이용하였다. 이 값은 영상의 종류나 필요에 따라 바꿀 수 있는 값으로, 필터링되는 정도를 조절할 수 있다.

노이즈를 제거하기 위해서 최대 세 장의 프레임에서 윈도우를 구성해 사용한다. 먼저, 현재 프레임 윈도우와 이전 프레임 윈도우 픽셀들의 차의 절대값의 합, SAD(sum of absolute differences)를 계산한다.

$$SAD1 = \sum_i \sum_j |g(i,j,t) - g(i,j,t-1)| \quad (1)$$

SAD1이 문턱값보다 크거나 같은 경우, 이 두 프레임

```

Foreach pixel g(x,y) in the image
Set 5X5 window of g(x,y)
Foreach pixel g(i,j) in the window
Calculate the absolute difference |g(x,y)-g(i,j)|
If (|g(x,y)-g(i,j)| ≤ THRESHOLD)
    sum += g(i,j)
    count++
g'(x,y) = sum/count
    
```

그림 1. 적응평균필터 알고리즘  
Fig. 1. Algorithm of the adaptive mean filter.

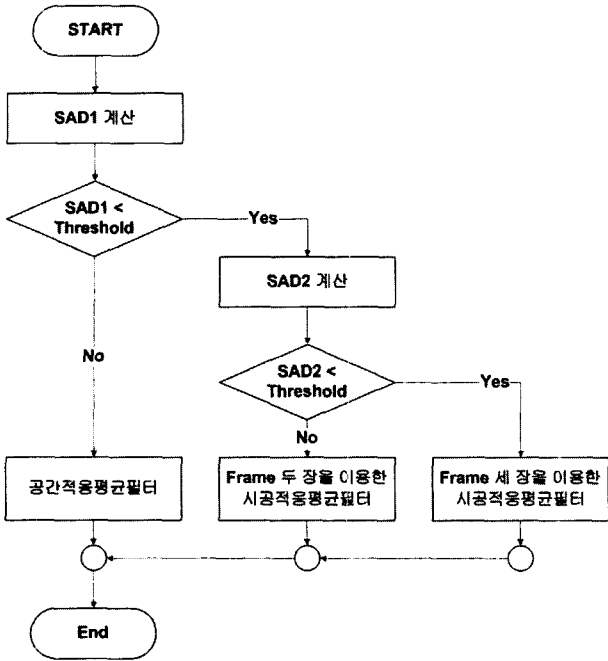


그림 2. 시공적응평균필터의 순서도  
Fig. 2. Flowchart of the spatio-temporal adaptive mean filter.

간에는 움직임이 있다고 판단하여 공간필터링을 한다. 움직임이 있는 경우에 시공필터링을 하면, 움직임이 노이즈로 인식되어 원치 않는 정보를 제거하기 때문이다. SAD1이 문턱값보다 작은 경우에는 현재 프레임 윈도우와 두 시간단위 이전의 프레임 윈도우 픽셀들의 차의 절대값의 합, SAD2를 계산한다.

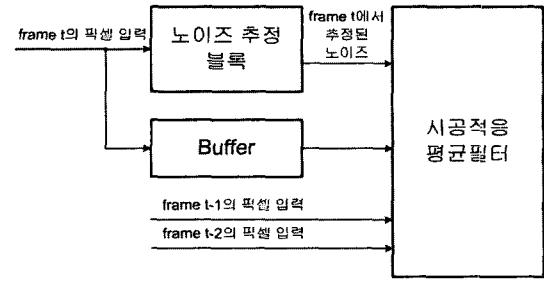
$$SAD2 = \sum_i \sum_j |g(i, j, t) - g(i, j, t - 2)| \quad (2)$$

SAD2 역시 문턱값과 비교하여, SAD2가 크거나 같으면 현재 프레임과 이전 프레임을 사용하여 시공적응평균필터링을 하고, 작으면 세 장의 프레임에 움직임이 없다고 판단, 세 장을 모두 사용한다.

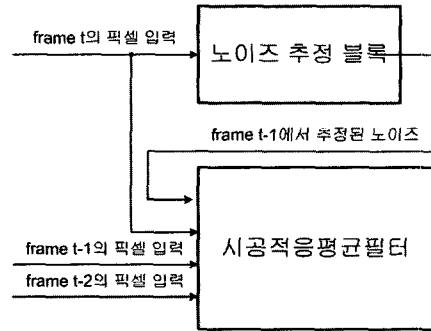
시공필터에서 사용되는 문턱값 역시 필요에 따라 조절할 수 있는 값이다. 본 논문에서는 상수 50으로 고정하여 필터링하였다.

### III. 하드웨어 구현을 위한 알고리즘 수정

영상처리 시스템의 구현에 가장 중요한 요소 중에 하나는 실시간 처리이다. 만약 시스템 내에 나눗셈이나 소수를 다루는 연산과 같이 복잡한 연산이 지나치게 많으면, 시스템의 속도를 감소시키고 궁극적으로는 실시간 처리를 어렵게 한다. 또한 외부 메모리로의 접근이



(a) 기존의 시공적응평균필터의 상위레벨 구조



(b) 제안하는 시공적응평균필터의 상위레벨 구조

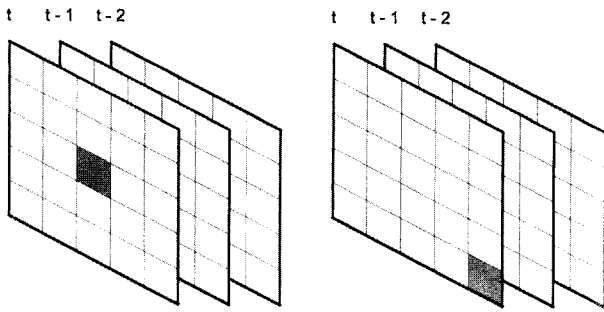
그림 3. 기존 알고리즘으로 설계한 구조와 하드웨어 구현 최적화를 위하여 수정한 알고리즘으로 설계한 구조

Fig. 3. Structure of the conventional algorithm and structure of the proposed algorithm which is modified for hardware optimization.

찾으면 접근하는 데에 걸리는 시간만큼 낭비하게 되어 시스템의 속도를 낮추는 요인이 된다.

설계한 시스템의 성능은 수치로 측정되기도 하지만, 결국 중요한 것은 영상을 보는 사람의 판단이다. 사람의 눈으로 인지할 수도 없는 성능의 개선을 위해서 속도가 느리고 면적이 큰 시스템을 구현하는 것은 무의미하다. 지나치게 복잡한 알고리즘은 피하여 하드웨어 복잡도를 줄이고 실시간 처리가 가능하도록 하는 것이 바람직하다.

적응필터는 본격적으로 픽셀을 필터링하기 전에 프레임의 노이즈 양을 추정하는 단계가 있다. 현재 프레임의 모든 픽셀 정보가 읽혀서, 노이즈 추정이 완료된 이후에 실제 필터링을 시작할 수 있다. 즉, 실시간 처리가 불가능하다. 문제는 이 뿐만이 아니다. 첫 번째 픽셀이 입력될 때부터 노이즈 추정이 끝날 때까지 입력되는 모든 정보들을 저장해야 하는 문제도 있다. 최근 처리해야 할 영상의 질이 높아지면서 하나의 프레임이 가지는 픽셀의 수가 크게 증가하고 있다. 예를 들어 일반적으로 LCD TV 혹은 모니터에 쓰이는 WXGA의 경우, 해상도가 1366x768로 한 프레임의 픽셀 수가 백만이 넘



(a) 기존의 필터링 윈도우 (b) 수정한 필터링 윈도우

그림 4. 하드웨어 구현을 고려한 필터링 윈도우 수정  
Fig. 4. Modification of the filtering window for hardware implementation.

는다. 이 픽셀들을 지역 메모리에 임시로 저장하는 것은 불가능하다. 외부 메모리에 저장할 수 있지만, 시스템의 속도를 더 악화시킨다.

본 논문에서는 알고리즘의 적용력 측면에서 약간의 희생을 감수하고 속도, 메모리 면적 등에서 유리한 구조를 제안한다. 그림 3.a에 기존의 알고리즘에 의한 구조를 나타내었고 3.b에는 알고리즘을 수정하여 노이즈 추정과 필터링이 병렬적으로 수행되도록 한 구조를 나타내었다. 여기서는 이전 프레임에서 추정한 노이즈를 현재 프레임의 필터링에 사용한다. 이는 연속된 두 프레임이 크게 다르지 않다는 가정 하에 가능하다. NTSC(National Television System Committee) 방식에서 프레임 수는 초당 30개로 현재 프레임과 직전의 프레임에 큰 차이가 없으므로, 본래의 알고리즘과 성능에서 큰 차이 없이 속도 측면에서 이득을 볼 수 있다.

그림 4에 또 다른 수정사항을 나타내었다. 필터링 단계에서는 그림 4.a와 같이 노이즈를 제거하고자 하는 픽셀이 중심이 되도록 하는 단위윈도우를 만들어 사용한다. 윈도우 내의 모든 픽셀의 정보를 가지고 있어야 필터링을 수행할 수 있다. 기존의 윈도우를 이용하면 단위윈도우가 완성되는 시점과 노이즈 제거 대상 픽셀이 입력되는 시점이 다르다. 이렇게 되면, Hsync, Vsync가 어긋나서 동기화를 위한 추가적인 연산이 필요하고 그에 따라 더 큰 버퍼링 공간이 필요하다. 제안하는 구조에서는 단위블록을 그림 4.b와 같이 구성하여 블록이 완성되는 시점과 대상 픽셀의 입력 시점을 일치시켰다.

IV. 제안하는 구조

그림 5에 제안하는 구조를 시스템 레벨에서 나타내

었다. 입력되는 픽셀은 RGB 각 요소가 8bits씩, 총 24bits을 가진다. 필터 내부에서 처리하는 정보의 양을 최소화하기 위해서, 내부에서는 RGB를 YCbCr로 바꾸어 사용한다. RGB가 필터 내부로 입력되면 가장 먼저 입력 제어단에서 이를 YCbCr로 바꾼다. Y는 계조와 관련된 정보를 가지고 있고, Cb와 Cr은 색깔과 관련된 정보를 가지고 있으므로, 제안하는 구조에서 필터링의 대상이 되는 것은 Y만이다. Y에 대한 필터링이 진행되는 동안 Cb와 Cr은 외부 SDRAM에 저장해둔다. 노이즈 추정과 필터링은 병렬적으로 진행되며 각 픽셀의 상황에 따라 적절한 알고리즘이 적용된다. Y의 노이즈 제거가 완료되면, 이에 해당하는 Cb, Cr이 외부 SDRAM에서 읽혀져 오고, 출력 제어단에서 YCbCr를 다시 RGB로 변환된 후, 최종적으로 출력된다.

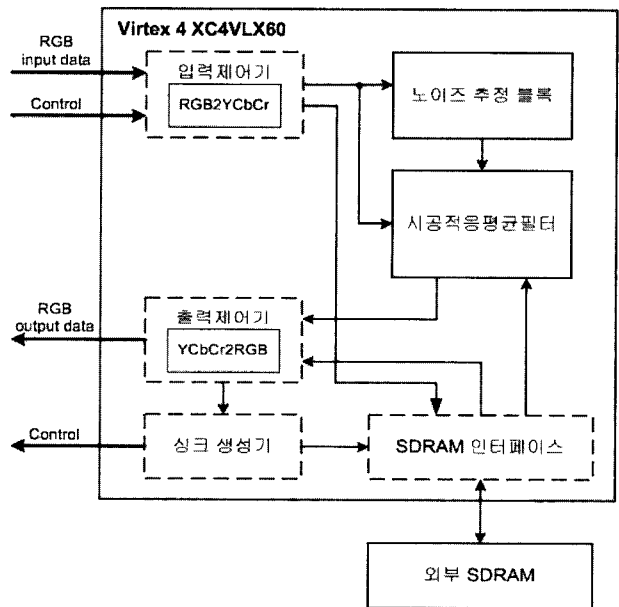


그림 5. 제안하는 시공적응평균필터의 시스템 레벨에서의 회로도

Fig. 5. System-level architecture of the proposed spatio-temporal adaptive mean filter.

1. 노이즈 추정 블록

제안하는 구조의 알고리즘에서 프레임의 노이즈는 프레임을 여러 개의 하위 윈도우로 나누어 각 윈도우의 표준편차를 구하고, 그 중 최소값으로 정해진다. 표준편차 계산에는 곱셈, 비교 연산뿐만 아니라 시간이 오래 걸리고 복잡한 제곱근 연산이 필요하다. 기존의 시공적응평균필터 알고리즘을 적용하면, 필터는 노이즈 추정이 끝나기를 기다렸다가 추정된 노이즈 값을 현재 프레임의 필터링에 사용한다. 따라서 노이즈 추정에 걸

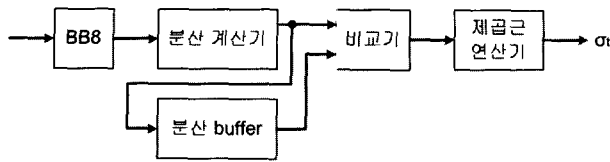


그림 6. 노이즈 추정 블록의 상위 레벨 회로도  
Fig. 6. High-level architecture of the noise estimation block.

리는 시간이 전체 시스템 속도에 직접적인 영향을 미치게 된다.

반면, 제안하는 구조에서는 이전 프레임에서 추정된 노이즈를 필터링의 기준으로 사용하면서, 노이즈 추정과 필터링이 병렬적으로 수행되게 하였으므로 노이즈 추정 블록의 레이턴시는 전체 시스템 속도에 영향을 미치지 않는다. 따라서 제공근 연산에 걸리는 시간에 관계없이 사용하기로 한다.

그림 6에 제안하는 노이즈 추정 블록의 구조를 나타내었다. 입력되는 픽셀들은 블록버퍼(BB8)에 임시로 저장되었다가 원하는 크기의 단위윈도우가 만들어지면 이를 알리는 신호와 함께 분산 계산 블록에 전달된다. 다음 단위윈도우가 완성될 때까지 이전에 계산한 분산값을 저장해두었다가 다음 윈도우의 분산과 비교하여 더 작은 값을 저장한다. 프레임 내의 모든 윈도우에 대한 분산 계산과 비교가 완료되면 마지막으로 제공근 연산을 하여 필터로 보낸다. 비교와 제공근 연산의 순서를 바꾸어 수행함으로써 복잡한 제공근 연산은 매 단위윈도우에 대해 수행하지 않고, 한 프레임에 대해 한 번만 수행하게 하였다.

## 2. 시공적응평균필터 블록

본 구조에서는 노이즈 제거에 최대 세 장의 프레임의 정보가 사용되므로, 시공적응평균필터의 내부는 그림 7에 보이는 것과 같이 세 개의 독립적인 필터로 구성되어 있다. 첫 번째 필터는 현재 입력되는 프레임(frame t)을 처리하는 일반적인 적응평균필터이다. 두 번째 필터와 세 번째 필터는 변형된 적응평균필터로서 외부 SDRAM(synchronous DRAM)으로부터 필요한 윈도우를 생성해서 과거 프레임 두 장(frame t-1, frame t-2)을 처리한다. 이 두 필터에서는 윈도우 내의 각 픽셀의 편차를 구할 때 노이즈를 제거하고자 하는 픽셀의 정보를 사용한다. 식 (3)은 필터2가 사용하는 결정과정을 나타낸다.

$$|g(x,y,t) - g(i,j,t-1)| < THRESHOLD \quad (3)$$

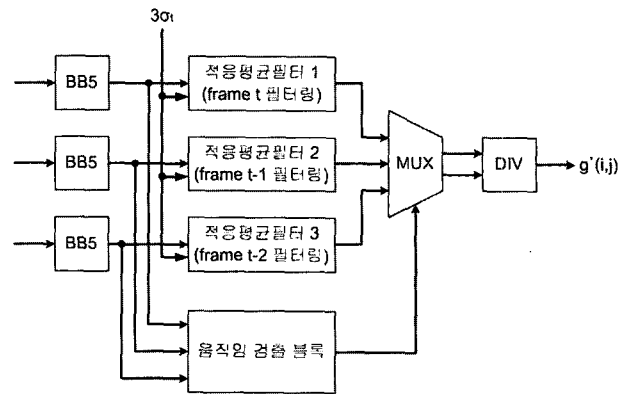


그림 7. 시공적응평균 필터의 상위 레벨 회로도  
Fig. 7. High-level architecture of the spatio-temporal adaptive mean filter block.

이들 필터들과 병렬적으로 움직임 검출 블록에서 SAD1과 SAD2가 계산된다. 이 값들에 따라 몇 장의 프레임이 노이즈 제거에 반영될 것인지 여부가 결정된다. 예를 들어 SAD1은 문턱값보다 작고 SAD2는 커서 프레임 두 장을 사용하는 것으로 결정하면, 필터1과 2에서 출력된 sum과 count를 합한다. 최종적으로 나눗셈기를 통과하면 노이즈가 제거된 픽셀이 출력된다.

이 때, 프레임의 맨 위쪽과 왼쪽의 일부 픽셀에 대해서는 온전한 단위윈도우를 구성할 수 없다는 문제가 있다. 이를 처리하는 방법으로는 가상의 데이터로 윈도우를 채워 넣거나, 존재하는 픽셀의 값을 복사하여 채워 넣는 방법이 있다. 그러나 두 방법 모두 일종의 가정에 근거한 것이므로 결과의 정확도를 보장하지도 않고, 게다가 구현은 지나치게 복잡하다. 따라서 온전한 윈도우를 구성할 수 없는 픽셀에 대해서는 필터링을 생략하고 그대로 출력한다. 본 논문에서 사용한 5×5 단위윈도우의 경우, 위쪽과 왼쪽의 네 줄 안에 위치한 픽셀들이 노이즈가 제거되지 않고 출력된다. 일반적인 HDTV의 해상도가 1366×768임을 고려할 때, 이는 육안으로 구별할 수 없는 미미한 부분으로 시스템의 성능에 영향을 미치지 않는다.

## V. 구현과 검증

제안하는 구조의 주요 블록들은 Verilog HDL로 설계하였고, 입력이나 출력을 제어하는 블록들만 VHDL로 설계하였다. Xilinx에서 제공하는 ISE 9.1.02i로 통합하여 합성하고 ModelSim 6.1f와 연동하여 시뮬레이션하였다. 이후에 Xilinx Virtex 4 Family 중의 하나인 XC4VLX60 위에 매핑하였다. 전체 슬라이스의 66%가

표 1. 제안하는 구조의 FPGA 구현 spec.  
Table 1. Specifications of the proposed system implementation.

	Used	Available	Utilization
CLB slices	17,697	26,624	66%
BRAMs	142	160	88%



(a) 노이즈가 추가된 원본 영상 (b) 노이즈를 제거한 영상  
그림 8. 검증에 사용된 입력영상과 출력영상  
Fig. 8. An Input image with noise and the filtered image.

사용되었고, BRAM(block RAM)은 88%가 사용되었다. 시스템의 속도는 최대 약 80MHz까지 가능하여 필터가 실시간으로 동작하기에 무리가 없다.

검증에는 임의의 HDTV급 동영상(해상도 1366X768)에서 연속적인 프레임을 캡처하여 사용하였다. 영상의 RGB를 텍스트 파일로 추출하여 테스트 벡터로 사용하였다. 한 프레임의 크기가 커서 시뮬레이션 시간이 지나치게 오래 걸리기 때문에 프레임 전체를 사용하지 않고, 1366X272 크기의 일부분을 사용하였다.

그림 8은 실험 영상의 일부를 확대한 것이다. 그림 8.a는 원본 영상에 10% 노이즈를 추가한 오염된 영상이다. 이 영상을 입력으로 하여 구현한 필터에 입력하여 얻은 결과 영상이 그림 8.b이다. 이 두 그림을 비교하면, 흰 배경 부분의 노이즈가 많이 제거된 것을 확인할 수 있다. 적응성을 가진 평균필터이므로 경계가 흐려지는 현상은 거의 없다.

## VI. 결 론

본 논문에서는 동영상의 가우시안 노이즈를 제거하는 시공간적응평균필터를 구현하였다. 효율적인 하드웨어 구현을 위하여 기존의 알고리즘은 부분적으로 수정되었다. 노이즈 추정 블록과 필터가 병렬적으로 동작하게 함으로써 시스템의 속도를 높여 실시간 처리가 가능하게 하였다. 필터링 윈도우의 형태를 변형하여 연산을

단순하게 하고 동기화를 쉽게 했다. 제안하는 구조는 FPGA 상에 구현하였고 전체 슬라이스의 66%가 사용되었다. 연속된 프레임의 RGB를 텍스트로 변환하여 테스트 벡터로 이용, 영상에서 노이즈가 제거되는 것을 확인하였다. 제안하는 구조는 LCD TV와 같이 고화질의 영상을 실시간으로 처리하는 시스템에 효율적으로 적용할 수 있다.

## 참 고 문 헌

- [1] S. V. Vaseghi, "Advanced digital signal processing and noise reduction." John Wiley & Sons Ltd., pp. 29, 2000.
- [2] R. C. Gonzales and R. E. Woods, "Digital Image Processing," Prentive-Hall, Inc., pp75-77, 2002.
- [3] B. E. Shi, "Spatio-temporal image filtering with cellular neural networks," in Proc. of IEEE International Conference on Neural Networks, pp. 1410-1415, 1996.
- [4] B. Smolka, M. Szczepanski, K. N. Plataniotis and A. N. Venetsanopoulos, "On the efficiency of random walk approach to noise reduction in color images," in Proc. of IEEE International Conference on Image Processing, pp. 138-141, Thessaloniki, Greece, October 2001.
- [5] H. Hisamitsu and K. Arakawa, "Time and spatial e-filter for improving image quality of compressed video," in Proc. IEEE International Midwest Symposium on Circuits and Systems, pp. 289-292, Hiroshima, Japan, July 2004.
- [6] I. Gasteratos, A. Gasteratos and I. Andreadis, "An Algorithm for Adaptive Mean Filtering and Its Hardware Implementation," Journal of VLSI Signal Processing, Vol. 44, pp. 63-78, August 2006.
- [7] S-W. Lee, V. Maik, J. Jang, J. Shin and J. Paik, "Noise-Adaptive Spatio-Temporal Filter for Real-Time Noise Removal in Low Light Level Images," IEEE Trans. Consumer Electronics, Vol. 5, Issue 2, pp648-653, May 2005.
- [8] D-H. Shin, R-H Park, S. Yang and J-H Jung, "Block-based noise estimation using adaptive Gaussian filtering," IEEE Transactions on Consumer Electronics, Vol. 51, No. 1, February 2005.

저 자 소 개



홍 혜 정(학생회원)  
 2006년 연세대학교 전기전자  
 공학과 학사 졸업.  
 2008년 현재 연세대학교 전기전자  
 공학과 석박통합과정.  
 <주관심분야: SoC 설계 및 응용,  
 테스트>



김 현 진(학생회원)  
 1997년 연세대학교 전기공학과  
 학사 졸업.  
 1999년 연세대학교 전기공학과  
 석사 졸업.  
 2005년 삼성전기 중앙연구소  
 선임연구원.

2008년 현재 연세대학교 전기전자공학과  
 박사과정.

<주관심분야: SoC 설계 및 응용, CAD>



강 성 호(정회원)  
 1986년 서울대학교 제어계측  
 공학과 학사 졸업.  
 1988년 The University of Texas,  
 Austin 전기 및 컴퓨터 공  
 학과 석사 졸업.  
 1992년 The University of Texas,  
 Austin 전기 및 컴퓨터  
 공학과 박사 졸업.

1992년 미국 Schlumberger 연구원.  
 1994년 Motorola 선임연구원.  
 2008년 현재 연세대학교 전기전자공학과 교수.  
 <주관심분야: SoC 설계 및 SoC 테스트 >