

# 양방향 모델을 적용한 Full-image Guided Filter의 효율적인 VLSI 구조

이겨레\*, 박태근<sup>o</sup>

## Efficient VLSI Architecture of Full-Image Guided Filter Based on Two-Pass Model

Gyeoree Lee\*, Taegeun Park<sup>o</sup>

요약

Full-image guided filter는 커널 윈도우 영역만 필터링에 반영되는 기존의 커널 윈도우 기반 가이드 필터와 달리 가중치 전파 도식과 양방향 모델이 적용되어 영상의 모든 픽셀이 필터링에 반영된다. 이로써 가이드 필터의 경계 보존과 평활화 등의 가이드 이미지 필터의 특성을 유지하면서도 계산 복잡도를 개선할 수 있다. 본 논문에서는 full-image guided filter의 더 빠른 처리가 필요한 스테레오 비전 및 각종 실시간 시스템 분야에 적용될 수 있도록 효율적인 하드웨어 구조를 제안하였다. 필터링 프로세스에서 발생하는 각종 데이터의 종속성 분석과 영상의 PSNR 분석, 데이터 빈도 분석 등을 통하여 적합한 하드웨어 구조를 제안하였다. 또한 양방향 모델이 적용된 가중치 연산 모듈의 휴식 구간이 최소화되도록 효율적인 스케줄링을 하였고 실시간 처리가 가능하게 하였다. 제안한 하드웨어 구조는 동부하이텍 0.11um 표준셀 라이브러리로 합성하였을 경우 최대 동작주파수 214MHz(384\*288 영상: 965 fps)와 76K(내부 메모리 제외) 게이트의 하드웨어 복잡도를 나타냈다.

**Key Words** : realtime system, guide filter, VLSI architecture, image filter, two-pass model

### ABSTRACT

Full-image guided filter reflects all pixels of image in filtering by using weight propagation and two-pass model, whereas the existing guide filter is processed based on the kernel window. Therefore the computational complexity can be improved while maintaining characteristics of guide filter, such as edge-preserving, smoothing, and so on. In this paper, we propose an efficient VLSI architecture for the full-image guided filter by analyzing the data dependency, the data frequency and the PSNR analysis of the image in order to achieve enough speed for various applications such as stereo vision, real-time systems, etc. In addition, the proposed efficient scheduling enables the realtime process by minimizing the idle period in weight computation. The proposed VLSI architecture shows 214MHz of maximum operating frequency (image size: 384\*288, 965 fps) and 76K of gates (internal memory excluded).

\* 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2011-0009513).

※ 본 연구는 2015년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음(M-2015-B0002-00059).

• First Author : Department of Information, Communication, and Electronic Engineering, The Catholic University of Korea, lkrkr02@gmail.com, 학생회원

o Corresponding Author : Department of Information, Communication, and Electronic Engineering, The Catholic University of Korea, parktg@catholic.ac.kr, 종신회원

논문번호 : KICS2016-09-237, Received September 2, 2016; Revised November 3, 2016; Accepted November 7, 2016

## I. 서 론

이미지 필터링은 컴퓨터 비전과 영상처리 등의 분야에서 영상의 복원, 흐림 및 선명 효과, 경계 감지, 특징 추출 등 다양한 역할을 한다. 최근에는 우수한 경계보존 특성을 갖는 양방향 필터(bilateral filter), 가이드 이미지 필터(guided image filter) 등이 각광받고 있다. 그 중에서 가이드 이미지 필터<sup>[1]</sup>는 가이드 이미지를 사용하여 영상을 평활화하면서도 경계를 잘 보존할 수 있고 디테일 강조, HDR 압축, 안개제거, 결합 업 샘플링 등의 특징을 지니고 있어서 스테레오 비전<sup>[2]</sup>, 영상 복원<sup>[3]</sup> 등 영상처리 관련 응용분야에 적용된다.

앞서 언급한 양방향 필터는 경계 보존 특성을 갖고 있지만 디테일 변질<sup>[4]</sup>과 HDR 압축<sup>[5]</sup>에서 화소 상 결함이 존재한다. 이는 해당 부분에서 필터링 가중치가 부적절하게 반영되는 물체 간 경계에서 주로 발생한다. 이러한 문제점을 해결하기 위해 제안된 필터가 가이드 이미지 필터다. 가이드 이미지 필터는 필터링을 수행할 때 가이드 이미지가 반영되어 물체의 경계 부분에서의 성능이 양방향 필터에 비하여 더 우수하다. 또한 가이드 필터는 양방향필터와 비슷한 계산 복잡도를 갖기 때문에 실시간 처리가 필요한 여러 분야에 쉽게 적용이 가능하다. 가이드 이미지 필터가 사용된 예로는 K. He 등의 알파 마스크 개선을 위한 후처리 단계에서의 가이드 필터 적용<sup>[6]</sup>, 칼라 구조물의 보존에서 색 노이즈로 인한 억제효과를 개선하기 위해 전송된 색의 평활화에 가이드 이미지 필터 적용<sup>[7]</sup>, Zhang 등의 비공된 전송 맵의 개선에 가이드 이미지 필터 적용<sup>[8]</sup> 등이 있다.

그러나 가이드 이미지 필터는 고정된 크기의 커널 윈도우를 사용하여 필터링을 수행하기 때문에 저질감 영역에서 필터링 성능이 좋지 않다. 또한 칼라 가이드 이미지 사용 시 선형 계산 복잡도를 보이더라도 실시간 처리에 한계가 있다. 이러한 단점은 가이드 이미지 필터를 여러 프레임에 대하여 적용해야 하는 스테레오 매칭 등의 시스템에 적용하기에 무리가 있다. 가이드 이미지 필터의 한계점 들을 개선하기 위하여 서브 샘플링기법을 이용하여 속도를 개선하는 연구<sup>[9]</sup>, 양방향 모델을 적용한 가이드 이미지 필터<sup>[10]</sup> 등이 제안되었다. full-image guided filter<sup>[10]</sup>는 저질감 영역에서의 경계 보존을 만족하며, 가중치를 분산해서 연산함으로써  $O(1)$ 의 계산 복잡도를 갖는다.

가이드 이미지 필터를 실시간 처리 시스템에 적용하기 위해서는 더 빠른 처리가 가능한 VLSI 구조로

설계할 필요가 있다. 관련 연구로 Kao 등이 30 frames/s 을 만족하는 가이드 필터의 VLSI 구조를 두 개의 통합 이미지 구조로써 설계한 연구<sup>[11]</sup>가 있으며 가이드 이미지 필터와 유사한 기능의 필터인 결합 양방향 필터(joint bilateral filter)를 Tseng 등이 메모리 감소 기법 등을 적용하여 설계한 연구<sup>[12]</sup>, Han 등의 고성능 스테레오 비전을 위한 양방향 필터의 VLSI 구조 설계<sup>[13]</sup> 등이 있다.

본 논문에서는 양방향(two-pass) 모델을 적용한 full-image guided filter<sup>[10]</sup> 알고리즘을 효율적인 VLSI 하드웨어 구조로 설계하였다. 필터링 과정에서 데이터 종속성을 분석하여 시간적으로 가중치 연산을 수평과 수직 연산 단계로 구분하였으며, 각 단계의 연산은 동일한 모듈에서 순차적으로 수행된다. 또한 제안된 시스템은 입력 신호들을 제어하는 제어 모듈과 가중치 연산을 수행하는 가중치 연산 모듈, 수평 연산의 결과를 수직 연산의 입력으로 사용 가능하게 저장하는 행렬변환 메모리(transpose memory), 정규화 및 나눗셈기로 구성되어 있다. 가중치 연산 모듈은 동시에 양방향 연산을 수행하기 위해 병렬로 구성된 순방향 모듈과 역방향 모듈을 포함한다. 해당 모듈에서는 처리과정에서의 휴식구간을 최소화 하고 중간 데이터가 저장되는 임시 저장 공간을 최소화 하도록 스케줄링 하여 구조를 설계하였다. 또한 필터링 과정 중 지수함수 연산 모듈은 빈도수 분석과 PSNR 분석을 통해 65단계의 LUT(look-up table)로 설계하여 지수함수의 근사화로 인한 정확도와 하드웨어 요구량을 개선하였다.

본 논문의 구성은 다음과 같다. II장에서는 full-image guided filter의 필터링 알고리즘을 설명하고, III장에서는 제안하는 하드웨어 구조와 스케줄링 방법에 대하여 설명한다. IV장에서는 하드웨어 구조를 합성한 결과를 여러 선행연구 결과들과 비교 분석하고 V장에서는 앞선 내용에 대한 결론을 맺는다.

## II. Full-image guided filter

그림 1은 경계가 잘 보존되는 영상 평활화 특성과 디테일 향상, HDR 압축, 안개제거 등의 효과를 보이는 가이드 이미지 필터링의 예를 보여준다.

본 논문에서 적용한 full-image guided filter는 입력 영상의 모든 요소가 가중치 전파에 의해 분산되어 연산된다. 이로서 필터링의 계산 복잡도를 획기적으로 낮출 수 있다<sup>[10]</sup>. full-image guided filter의 필터링을 과정을 설명하면 다음과 같다.



그림 1. 가이드 필터의 경계 보존 평활화 특성  
Fig. 1. Edge-preserving smoothing property of guide filter

$$C'_i = \frac{1}{N_i} \sum_{j \in \Omega} W_{i,j}(I) C_j \quad (1)$$

위 수식(1)에서  $C_j$ 는 필터링 입력이며 특정 신호나 영상의 픽셀 데이터가 될 수 있다. 이 때,  $j$ 는 전체 입력 데이터 중에서 필터링할 영상 픽셀의 위치인  $i$ 에 대하여 필터링에 필요한 입력데이터의 위치를 나타낸다.  $j$ 의 범위는 일반적인 가이드 필터에서는 커널 윈도우의 영역이지만 full-image guided filter에서는 전체 영상의 영역이다.  $\sum_{j \in \Omega} W_{i,j}(I)$ 는 필터링 입력  $C_j$ 와 곱해짐으로써 가이드 이미지  $I$ 에서  $i$  위치에 대한 필터링을 수행하는 가중치 누적 값이다.  $N_i$ 는 정규화 상수로써 모든 픽셀  $j$ 에 대한 가중치  $W_{i,j}$ 의 누적 합이며 실제 연산에서는  $N_i = \sum_{j \in \Omega} W_{i,j}(I)$  로 값을 구할 수 있다.

$$W_{i,j}(I) = \prod_{(p,q) \in P_{i,j}} T_{p,q}(I), \quad (2)$$

$$T_{p,q}(I) = \exp\left(-\frac{\|I_p - I_q\|}{\sigma}\right), \quad (3)$$

식(2)는 식(1)의 가중치를 연산하는 부분으로 여기서 가중치는 가이드 이미지의 픽셀 유사도를 누적하여 곱한 것이다. 식(3)에 정의된 픽셀 유사도  $T_{p,q}$ 는 인접한 두 픽셀  $p, q$ 의 픽셀 값을  $I_p, I_q$ 라 하였을 때 픽셀 값 간의 유클리드 거리와 지수함수를 이용하여 구할 수 있다. 인접한 두 픽셀  $p, q$ 는 픽셀  $i$ 와 픽셀  $j$ 사이의 경로에 있는 인접한 픽셀이며  $\sigma$ 는 필터링의 정도를 조절하는 파라미터이다.

그림 2에서  $(i, j)$  사이의 경로는 무수히 많기 때문에 계산 복잡도 개선을 위하여 수평우선 방식을 채택하여 경로를 단순화했다. 식(2)로 가중치를 계산할 때

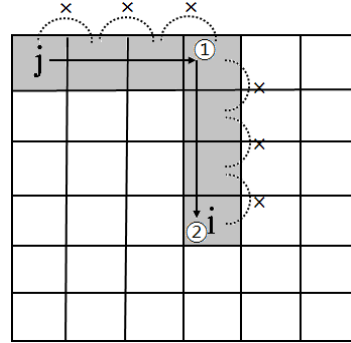


그림 2. 가중치 연산 방식  
Fig. 2. Weight calculation method

$j$  위치에서 가중치를 1로 초기화한 후  $i$ 가 있는 열까지 수평 연산한 뒤, 수직 계산하여  $(i, j)$ 간 최종적인 가중치를 구한다. 경로를 단순화하여 계산 복잡도가 줄었음에도 필터링 기준 위치에 대해 다른 위치 간 가중치를 연산에 사용되는 성분들이 많이 중복되므로 양방향(two-pass) 모델을 적용하여 계산 복잡도를 크게 개선할 수 있다.

$$\begin{aligned} C_r^{imed} &= \sum_{i=0}^{r-1} W_{i,r}(I) C_i + \sum_{i=r+1}^n W_{i,r}(I) C_i + C_r \\ &\quad \text{순방향} \qquad \qquad \qquad \text{역방향} \\ &= \sum_{i=0}^{r-1} \left( \prod_{u=i+1}^r T_{u,-}(I) \right) C_i \\ &\quad + \sum_{i=r+1}^n \left( \prod_{u=r}^{i-1} T_{u,+}(I) \right) C_i + C_r \end{aligned} \quad (4)$$

식(4)는 양방향 모델의 적용을 통해 가중치가 분산되어 연산되고 그 결과는 식(1)의 가이드 필터링의 결과와 같음을 보여준다. 양방향 모델은 가중치를 누적시키는  $\sum_{j \in \Omega} W_{i,j}(I) C_j$  연산을 행 방향과 열 방향에 대하여 나누고 각 방향의 연산 결과를 가중 중간 합( $C_r^{imed}$ )으로 정의한 후 식(4)의 수평 성분과 수직 성분에 대해 연산을 진행하면 필터링 된 값을 얻을 수 있다. 가중 중간 합  $C_r^{imed}$ 는 입력 영상과의 곱이 함께 반영된 필터링의 중간 값이다. 식(4)에서  $u^-$ 는 식(2)의 픽셀 유사도를 구할 때 식(3)에 의해 누적 곱이 수행되는 위치를 표현하는 지시자이다.  $u^-$ 는 위치  $u$ 의 인접하는 왼쪽 픽셀이며  $u^+$ 는  $u$ 의 오른쪽 픽셀을 나타낸다.

$$\begin{aligned}
 A_r^L &= C_r + \sum_{i=0}^{r-1} \left( \prod_{u=i+1}^r T_{u-,u}(\hat{I}) \right) C_i \\
 &= C_r + T_{r-1,r}(C_{r-1} + T_{r-2,r-1}(C_{r-2} + \\
 &\quad \dots + T_{0,1}(C_0 \dots))) \\
 &= C_r + T_{r-1,r}A_{r-1}^L
 \end{aligned} \tag{5}$$

식(4)를 풀어서 정리하면 식(5)와 같이 나타낼 수 있으며 오른 쪽 방향에서도 동일한 방식으로 연산이 가능하고 수직 성분에도 같은 방식으로 연산을 할 수 있다.

$$C_r^{imed} = A_r^L + A_r^R - C_r \tag{6}$$

수평 우선 경로 선정 방식과 양방향 모델의 적용을 통하여 구한 분산된 중간 가중 합은 식(6)의 간단한 덧셈으로 합쳐질 수 있다. 같은 방식으로 수직 연산에 대하여 식(5)와 식(6)의 연산을 하면 최종 필터링 결과를 얻게 된다. 위의 정리를 통해 얻은 양방향 알고리즘은 식(1)에서 설명한 full-image guided filter의 계산 복잡도를 O(1)로 크게 낮춘다. 특정 위치 r에 대한 중간 가중 합을 구할 때 수평 성분에 대한 계산에서 분산된 중간 가중 합의 성분들은 식(5)에 의해 1번의 곱셈과 1번의 덧셈으로써 구해지며, 이 성분들은 식(6)에서 양방향에서 얻어지는 중간 가중 합인  $A_r^L$ ,  $A_r^R$  로 표현된다. 두 성분은 같은 방식으로 연산되므로  $C_r^{imed}$  는 2번의 곱셈과 4번의 덧셈으로 연산할 수 있다. 최종 필터링 결과를 얻기 위해  $C_r^{imed}$  를 식(1)에서의  $C_j$  로 사용하여 같은 방식으로 수직 연산을 하게 되면 총 4번의 곱셈과 8번의 덧셈으로 필터링 프로세스에 맞게 수행된 결과를 얻을 수 있다. 정규화 상수  $N_i$  도 같은 연산 방식으로 진행되므로 최종 필터링에 필요한 연산은 8번의 곱셈과 16번의 덧셈 그리고 정규화 과정에 의한 1번의 나눗셈을 통하여 full image guided filter의 필터링을 처리할 수 있다.

### III. 제안하는 구조

양방향 모델을 적용한 full-image guided filter는 가중치가 분산되어 연산되므로 불필요한 중복 연산을 피할 수 있고 단순한 계산 요소들의 조합으로 필터링 할 수 있다. 본 논문에서는 이러한 필터를 더욱 빠른 처리가 필요한 스테레오 비전 및 실시간 응용 시스템 분야에서 적용 가능한 하드웨어 구조를 제안한다.

#### 3.1 제안하는 풀이미지 가이드 필터의 구조

그림 3은 양방향 모델을 적용한 full-image guided filter의 전체적인 하드웨어 구조를 나타낸다. 제안된 시스템은 가중치를 계산하는 가중치 연산 모듈(weight calculation)과 중간 가중치 결과를 저장하는 메모리(transpose memory), 각 모듈의 입력신호를 제어하는 모듈(control module) 그리고 정규화 모듈과 최종 필터링 값을 계산하는 나눗셈기(DIV)로 구성되어 있다.

데이터의 흐름과 각 구조를 설명하면 다음과 같다. 그림 3에서 제안된 시스템에서 필터링은 수평과 수직 가중치 연산을 순차적으로 두 번 수행한다. 수평 연산을 진행할 때, 행의 양 끝에서부터 분산된 가중치 연산을 진행하고 양 방향의 두 연산 결과를 이용하여 식(6)과 같이 가중치의 수평 성분에 대한 중간 계산 값이 연산된다. 제어모듈의 제어 때문에 가중치 연산모듈로 들어가는 가이드 이미지 데이터는 양 방향에 대해 각 2개씩 4개이다. 그림 3에서는 쉬운 이해를 위하여 합쳐서 표기하였다. 필터링의 수직 연산은 데이터 의존성 때문에 수평 성분 연산 결과를 행렬변환 메모리(transpose memory)에 저장한 후 수직 성분 연산을 수행할 때 다시 입력으로 사용한다. 양방향 가이드 이미지 필터는 정규화 상수인  $N_i$  가 필요 하다. 그림 3에서 정규화 상수 연산 모듈과 그 결과를 저장하기 위한 메모리가 별도로 필요하다. 그리고 나눗셈기(DIV)에서는 가중치 연산 모듈의 값과 정규화 상수 모듈의 값을 나누어 줌으로써 필터링 결과를 0 - 255의 범위로 제한한다. 본 논문에서는 비복원 어레이 나눗셈기(non-restoring array divider) 구조를 적용하였다<sup>[4]</sup>.

그림 4(a)는 가중치 연산 모듈의 구조를 나타낸다. 이 모듈에서는 식(6)의 연산이 효율적인 스케줄링에 의해 수행된다. 이 때 수평(수직) 연산의 최/우(상/하)

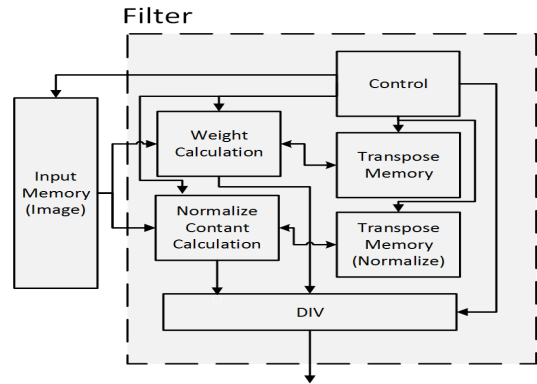
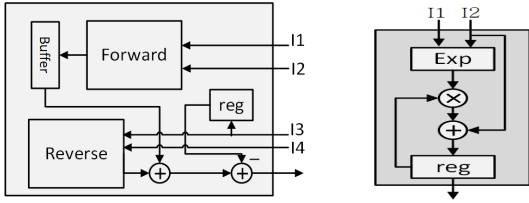


그림 3. 제안한 full-image guided filter 구조  
Fig. 3. Proposed full-image guided filter structure



(a) 가중치 연산 모듈 (b) 순방향(역방향) 모듈  
 그림 4. 가중치 연산 모듈과 순방향(역방향) 모듈  
 Fig. 4. Weight calculation module and forward (reverse) module

방향의 처리는 순방향(forward) 모듈과 역방향(reverse) 모듈에서 진행된다. 식(6)에서는 수평 연산을 기준으로 양방향을 나타내는  $A_r^L$ 과  $A_r^R$ 의 기호로 표현되었다. 그림 3에서 언급한 가중치 연산 모듈로 입력되는 4개의 가이드 이미지 데이터가 가중치 연산 모듈로 들어오면 각 2개씩 나뉘어 순방향, 역방향 모듈로 들어가서 분산된 가중치를 구하는 연산을 수행한다. 순방향 모듈에서 계산되는 부분 연산의 결과는 스케줄링에 의해 버퍼에 저장되었다가 역방향 모듈에서 부분 연산의 결과가 나올 때마다 조합되어 가중치 연산 모듈의 출력으로 나가게 된다. 스케줄링에 대한 설명은 뒤에 다시 언급하겠다. 가중치 연산 모듈에서의 비트 폭은 PSNR 분석을 통해 순방향(역방향) 모듈에서 24비트로 설정하였다. 가중치의 특성상 누적 연산이 반복되어 값이 커지기 때문에 연산이 수행될 때 결과 값들의 최대치가 오버플로우 되지 않도록 정수 비트 폭을 설정하였다. 그림 4(b)는 분산된 가중치를 누적하는 순방향(역방향) 모듈의 구조이다. 순방향과 역방향 모듈은 동시에 연산을 수행하기 위해 두 개로 구성되었을 뿐 같은 구조를 갖는다. 이 모듈에서는 가이드 이미지 데이터 I1, I2, I3, I4가 필터링 식(3)에 맞춰 양방향으로 분산된 가중치를 연산한다.

본 논문에서는 가이드 이미지와 입력 이미지를 같은 흑백 영상으로 적용하였기 때문에  $\|I_p - I_q\|$ 가 1차원이 되어 그 차로서 손쉽게 구현할 수 있다. 지수함수의 입력에 대한 빈도 분석결과 가능한 입력 값의 범위는 0 ~ 255이지만 대부분 작은 값에 몰려있다. 따라서 본 논문에서는 빈도수가 작은 구간은 단일 값으로 할당하여 지수함수를 위한 LUT(look-up table)의 단계를 줄여 구조를 단순화하였다. 그림 5에서 LUT 단계와 PSNR 변화를 통해 빈도수가 65단계 이상일 경우엔 PSNR이 크게 감소하지 않음을 확인할 수 있다. 이를 바탕으로 LUT를 65단계로 구성하였다. LUT의 각 요소에 대한 비트 폭은 PSNR 분석을 통해 목표로 하는 45dB 이상의 품질을 유지하도록 9비트

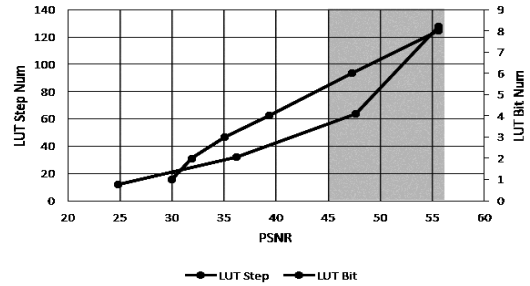


그림 5. PSNR 분석을 통한 지수함수 구현  
 Fig. 5. PSNR analysis of exp(x)

로 설정하였다. 지수함수의 출력 데이터는 이미 연산되어 레지스터에 저장된 분산 가중치( $A_r^{L-1}$ )와 곱해져 현재 위치의 분산 가중치( $A_r^L$ )로 연산된다. 이 때, 레지스터에 저장되는 데이터는 분산 가중치가 누적되기 때문에 오버플로우를 고려해 23비트로 선정하였다. 곱셈 연산을 위해  $23 \times 9$ 의 어레이 곱셈기(unsigned array multiplier)<sup>[14]</sup>를 설계하였다.

### 3.2 스케줄링과 데이터 재사용

수평연산을 진행할 때 영상의 각 행에 대해 분산된 가중치연산은 행의 양 끝에서부터 시작한다. 식(5)의 결과인  $A_r^L$ ,  $A_r^R$ 을 구하는 과정을 분석해 보면  $A_r^L$ ,  $A_r^R$ 는 서로의 데이터 의존성 때문에 식(6)은 동시에 연산될 수 없다. 따라서 본 논문에서는  $A_r^L$ ,  $A_r^R$ 을 순차적으로 연산하였으며 이로 인한 연산의 비효율성을 개선하기 위하여 파이프라인 스케줄링을 적용하였다.

그림 6에서와 같이 처음에는 한 행에 대한 순방향 연산( $0^{th} A_r^L$ )만 진행하며 이때 데이터 의존성 때문에 역방향 모듈은 휴식하게 된다. 그리고 계산된 순방향 모듈의 출력은 버퍼에 저장되었다가 그 다음 행이 연산될 때 출력되어 동일한 행에 대한 역방향 모듈의 출력( $0^{th} A_r^R$ )과 식(6)의 계산식대로 가중치를 연산하게 된다. 첫 행에 대하여 이와 같은 방식으로 연산이 진행된 이후에는 순방향과 역방향 모듈들은 그림 6에서

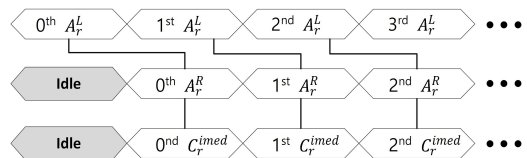


그림 6. 가중치 모듈의 스케줄링  
 Fig. 6. Scheduling of weight calculation module



와 같이 휴식 없이 가중치를 구하게 된다.

이 때 버퍼의 크기는 순(역) 방향 모듈의 출력 데이터의 폭을 갖고며 영상의 행(열) 길이만큼의 크기와 갖는다. 데이터 의존성에 의해 역방향 모듈 출력 데이터가 준비되는 시점까지 버퍼에 이전 행에 대한 순방향 모듈의 출력 결과를 저장하고 있어야 한다. 그래서 단순하게는 행(열)크기×2 의 길이를 갖는 버퍼를 가지고 있어야 하며 이는 비효율적이다. 그러므로 본 논문에서는 버퍼의 저장 위치를 고정하지 않고 버퍼에서 데이터의 출력지점에 순방향 모듈의 출력인 새로운 데이터를 저장하도록 구성하였다.

스케줄링 방식에 대한 쉬운 이해를 위해 그림 7과 표 1을 통하여 4×4 크기의 작은 영상에 대해 순방향, 역방향 모듈의 연산 수행 위치와 순방향 모듈의 출력에 대한 버퍼상태를 나타내었다. 순방향 모듈은 첫 행(A, B, C, D)에 대해 앞서 그림 6에서 설명한 바와 같이 연산을 수행하고 이때 역방향 모듈은 휴식 한다. 이 때 A 위치에 대한 순방향 출력은 버퍼에 저장된다. 이어서 순방향 모듈은 다음 픽셀에 대한 데이터를 입력으로 하여 B 위치의 분산된 가중치 연산을 수행하고 결과는 A에 대한 순방향 출력을 저장한 위치의 다음 위치에 저장한다. 이어서 같은 행인 C, D에 대해서도 같은 연산과 저장을 반복하고 첫 행에 대한 연산이 끝난다.

순방향 모듈은 이어지는 두 번째 행(E, F, G, H)에 대하여 연산을 수행하고 이 때, 역방향 모듈은 첫 행에 대한 연산을 시작한다. 두 번째 행인 E, F, G, H 위치에 대해 순방향 모듈이 E 위치를 연산할 때 역방향 모듈은 첫 행의 D 위치에 대해 연산한다. 역방향 모듈의 연산이 끝나 결과가 출력됨과 동시에 버퍼에서 D를 읽어서 연산 후 가중치 모듈의 출력을 생성한다. 순방향 출력은 D 위치의 분산된 가중치가 저장되어 있었던 버퍼의 공간에 새로운 E 위치의 출력을 저장하게 된다. 그 후 순방향 모듈은 F 위치의 분산된 가중치를 연산하고 역방향 모듈은 C 위치의 분산된 가중치를 연산하며 마찬가지로 역방향 모듈의 출력이 나옴과 동시에 버퍼에서 C 위치의 분산된 가중치가 저장되어 있던 데이터를 내보내고 순방향 모듈의 새로운 F 위치에 대한 분산된 가중치를 해당 위치의 버퍼에 저장하게 된다. 이러한 방식으로 버퍼는 한 행마다 저장순서를 바꿔가며 순방향 모듈의 출력 데이터를 저장하였다가 역방향 모듈의 출력과 함께 연산하여 가중치가 연산된다. 수직 연산에 대해서도 동일한 스케줄링으로 필터링이 진행된다.

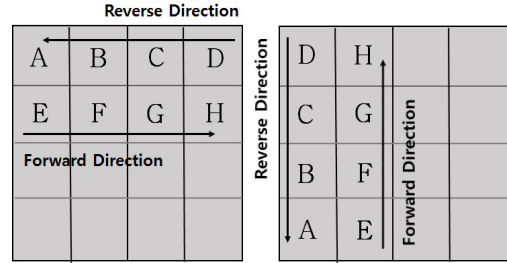


그림 7. 가중치 연산 방향(가로/세로)  
Fig. 7. Weight calculation direction

표 1. 가중치 모듈의 스케줄링 예시  
Table 1. Scheduling example of weight calculation

clock	1	2	3	4	5	6	7	8
Buffer (Store)				D	E	E	E	E
Forward Output)		B	B	B	B	B	G	G
Forward	A	A	A	A	A	A	A	H
Reverse					D	C	B	A
Module Output					D	C	B	A

#### IV. 설계 및 결과 분석

제안하는 full-image guided filter는 VerilogHDL로 설계하고 Active-HDL을 통해 검증하였다. 식(3)에서 필터링 정도를 조절하는 파라미터인  $\sigma$ 는 시뮬레이션을 통하여 가이드 필터의 특성 중 경계보존 평활화의 특성이 잘 들어나도록 16.0으로 설정하였다. 이를 바탕으로 설계한 VLSI 구조는 데이터의 비트할당과 비선형 함수의 근사화로 인하여 알고리즘으로 구현된 full-image guided filter의 결과와 PSNR 분석을 하였을 경우 약 45.6dB의 차이를 보였다. 제안한 구조를 동부 하이텍 110nm 표준셀 라이브러리를 사용하여 Synopsys Design Compiler로 합성하였을 때 표 2와 같은 합성 결과를 보였다. 제안하는 구조에서는 384×288 크기의 영상에 대하여 214MHz 동작 주파수와 초당 965 프레임의 월등한 동작 속도를 보였으며 메모리를 제외한 전체 시스템은 76K의 게이트 수를 나타내었다. 비슷한 크기인 320×240의 영상에 대해 양방향 필터의 VLSI 구조를 설계한 Han<sup>[11]</sup>의 구조와 비교하면 초당 프레임 수, 동작 주파수 등 모든 면에서 우수한 성능을 보임을 확인할 수 있었다. 제안한 구조에서 초당 처리 픽셀 수를 나타내는 처리율(throughput)은 106M(pixels/s)로서 비교 연구 결과와 비교하였을 때 우수한 성능을 보였다. 동일한 비교를

표 2. 다른 선행연구들과 결과 비교  
Table 2. Comparison with other ASIC filter Implementations

	Kao[9]	Tseng[10]	Han[11]	Proposed
Filter	guided	joint bilateral	bilateral	full-image guided
Technology	TSMC 90nm	UMC 90nm	TSMC 180nm	Dongbu hightech 110nm
Image size	1920×1080	1920×1080	320×240	384×288 (1920×1080)
Frame per second	30	30	144	965 (51)
Kernel window	31×31	31×31	11×11	full-image
Max. operating frequency	100MHz	100MHz	60MHz	214MHz
Throughput (pixels/sec)	62M	62M	11M	106M
Gate count	92.9K	276.2K	355K	76K

위하여 제안한 구조의 동작 주파수(214MHz)로 동일한 입력 영상 크기(1920×1080)에 대하여 필터링을 수행한다고 가정하였을 때 표 2에서 우측 괄호로 나타낸 결과와 같이 약 51fps로 속도 면에서 우수한 결과를 나타내었다.

### V. 결 론

본 논문에서는 양방향 모델을 적용한 full-image guided filter의 효율적인 VLSI 구조를 제안하였고 데이터 중속성 분석을 통한 가중치 연산 모듈의 재활용과 빈도수 분석을 통해 제안하는 지수함수의 LUT 구조, 가중치 연산 모듈의 휴식을 최소화하도록 하는 스케줄링 등을 적용한 구조는 실시간 처리를 가능하게 하는 빠른 처리 속도를 보여준다. 제안하는 구조를 동부하이텍 110nm 표준셀 라이브러리를 사용하여 합성하였을 때, 384×288 크기의 영상에 대하여 214MHz 동작 주파수와 초당 965 프레임의 동작 속도를 보였으며 메모리를 제외한 전체 시스템은 76K개의 게이트 수를 나타내었다.

### 감사의 글

저자들은 본 연구를 위하여 설계 소프트웨어를 제공하여 준 IDEC(IC Design Education Center)에 감사드린다.

### References

[1] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 35, no. 6, pp. 1397-1409, 2013.

[2] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother, "Real-time local stereo matching using guided image filtering," *IEEE Int. Conf. Multimedia and Expo*, pp. 1-6, 2011.

[3] J. Wasza, S. Bauer, and J. Hornegger, "Real-time preprocessing for dense 3D range imaging on the GPU: Defect interpolation, bilateral temporal averaging and guided filtering," *IEEE Workshop on Consumer Depth Cameras for Computer Vision*, pp. 1221-1227, 2011.

[4] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multiscale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 67:1 - 67:10, 2008.

[5] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257-266, 2002.

[6] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun, "A global sampling method for alpha matting," *IEEE Conf. CVPR*, pp. 2049-2056, 2011.

[7] A. Chia, S. Zhuo, R. K. Gupta, Y. Tai, S. Cho, P. Tan, and S. Lin, "Semantic colorization with internet images," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 156:1-156:8, 2011.

[8] J. Zhang, L. Li, Y. Zhang, G. Yang, X. Cao, and J. Sun, "Video dehazing with spatial and temporal coherence," *Vis. Comput.*, vol. 27,

pp. 749-757, 2011.

- [9] K. He and J. Sun, *Fast guided filter*, Tech. Report, arXiv:1505.00996v1, 2015.
- [10] Q. Yang, D. Li, L. Wang, and M. Zhang, "Full-image guided image filtering for fast stereo matching," *IEEE Sign. Process. Lett.*, vol. 20, no. 3, pp. 237-240, 2013.
- [11] C. Kao, J. Lai, and S. Chien, "VLSI architecture design of guided filter for 30 frames/s full-HD video," *IEEE Trans. CAS for Video Technol.*, vol. 24, No. 3, pp. 513-524, 2014.
- [12] Y. Tseng, P. Hsu, and T. Chang, "A 124 Mpixels/s VLSI design for histogram-based joint bilateral filtering," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3231-3241, 2011.
- [13] S. Han, "An architecture for high-throughput and improved-quality stereo vision processor," M.S. Thesis, Dept. Electr. Eng., Univ. of Maryland, College Park, MD, USA, 2010.
- [14] K. Hwang, *Computer arithmetic: principles, architecture, and design*, Wiley, 1979.

이겨레 (Gyeore Lee)



2016년 8월 : 가톨릭대학교 정보통신전자공학부 학사  
<관심분야> VLSI 설계, 영상처리

박태근 (Taegeun Park)



1985년 : 연세대학교 전자공학 석사  
1988년 : Syracuse Univ. 컴퓨터공학 석사  
1993년 : Syracuse Univ. 컴퓨터공학 박사

1991년~1993년 : Coherent Research Inc. VLSI 설계 엔지니어  
1994년~1998년 : 현대전자 System IC 연구소 책임연구원  
1998년~현재 : 가톨릭대학교 정보통신전자공학부 교수  
<관심분야> VLSI 설계, CAD, 컴퓨터 구조