

# 스테레오 영상으로부터의 실시간 변이정보 획득 알고리즘 및 하드웨어 구현

정회원 신완수\*, 최현준\*, 종신회원 서영호\*, 김동욱\*

## Real-time Disparity Acquisition Algorithm from Stereoscopic Image and its Hardware Implementation

Wan-Soo Shin, Hyun-Jun Choi *Regular Members*,  
Young-Ho Seo, and Dong-Wook Kim\* *Lifelong Members*

### 요약

본 논문에서는 기존의 변이 영상 획득 방법들에 비하여 시간 대비 정확도가 우수한 기법을 제안하고 H/W로 구현한다. 제안한 기법은 고속 연산이 가능한 화소 대 화소의 움직임 추정 기법을 이용한다. 움직임 추정 기법은 영상 내 텍스처의 분포 특성과 무관하게 정합 윈도우의 유사성에만 의존하기 때문에 추출된 변이정보의 정확도가 떨어진다. 이를 해결하기 위해서 영상의 국부 특성에 따른 가변 크기 윈도우 정합 기법을 도입하고, 영상 내 텍스처가 균일한 부분 및 물체의 윤곽선 부분에서도 높은 정확도를 얻는다.

제안한 기법은 고속 연산이 가능하도록 수행속도에 최적화된 하드웨어로 설계된다. 하드웨어는 Verilog-HDL로 설계하였고, Hynix 0.35 $\mu$ m CMOS 라이브러리를 사용하여 게이트수준으로 합성하였다. 구현한 하드웨어는 최대 120MHz의 클럭 주파수에서 초당 15 프레임을 안정적으로 처리할 수 있었다.

**Key Words** : stereo matching, disparity, hardware design, multi-view, 3D

### ABSTRACT

In this paper, the existing disparity acquisition algorithms were analyzed, on the bases of which a disparity generation technique that is superior in accuracy to the generation time was proposed. Basically it uses a pixel-by-pixel motion estimation technique. It has a merit of possibility of a high-speed operation. But the motion estimation technique has a disadvantage of lower accuracy because it depends on the similarity of the matching window regardless of the distribution characteristics of the texture in an image. Therefore, an enhanced technique to increase the accuracy of the disparity is required. This paper introduced a variable-sized window matching technique for this requirement. By the proposed technique, high accuracies could be obtained at the homogeneous regions and the object edges. A hardware to generate disparity image was designed, which was optimized to the processing speed so that a high throughput is possible. The hardware was designed by Verilog-HDL and synthesized using Hynix 0.35 $\mu$ m CMOS cell library. The designed hardware was operated stably at 120MHz using Cadence NC-VerilogTM and could process 15 frames per second at this clock frequency.

※ 이 논문은 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 수행된 연구임.  
(KRF-2008-313-D00735)

\* 광운대학교 실감미디어 연구소 (dwkim@kw.ac.kr)

논문번호 : KICS2009-03-090, 접수일자 : 2009년 3월 4일, 최종논문접수일자 : 2009년 11월 6일

## I. 서 론

차세대 비디오 서비스로 주목되고 있는 다시점(multi-view point) 및 자유시점(free-view point) 비디오 서비스는 사용자의 요구에 따라 새로운 가상시점을 제공한다. JVT(joint video team)에서는 MVD(multi-view video plus depth)를 이용한 자유시점 TV(free viewpoint TV, FTV)의 구조를 연구 중이다<sup>[1]</sup>. 동영상 압축코덱 중 가장 우수한 성능을 보이는 H.264/AVC<sup>[2]</sup>를 사용하여 여러 시점에서 획득한 영상을 동시에 압축하고 깊이정보와 함께 전송한다. 수신단에서는 압축된 원 영상과 깊이 정보를 복원한다. 또한 이 두 개의 정보를 렌더링(rendering)하여 임의의 시점에서 촬영된 영상과 같은 가상영상을 제공한다.

여기에 사용된 깊이정보는 공간상에서 카메라 시점을 기준으로 대상 물체까지의 거리로 정의된다. 깊이정보 추출의 기본원리는 각각의 눈에서 따로 관찰되는 물체의 상을 대뇌에서 분석 및 종합하여 물체의 원근을 인지하는 인간 시각 시스템(human visual system, HVS)에 근거를 두고 있다. 망막에 맺힌 각각 다른 물체의 상을 공학적으로 해석한 것이 스테레오 영상(stereoscopic image)이다.

깊이정보는 물체의 거리를 획득하는 카메라를 이용하여 능동적인 방법으로 획득할 수도 있지만 2개 이상의 카메라로부터 획득된 스테레오 영상의 기하관계를 이용하여 수동적인 방법으로 획득할 수도 있다. 깊이정보 추출에 대한 연구는 컴퓨터 비전 및 패턴인식 분야에서 오랜 기간 동안 연구되어왔다<sup>[3]</sup>.

상기와 같은 다양한 연구결과에서 보인 것과 같이 스테레오 영상에서 깊이정보를 추출하는 것은 대응성 문제(correspondence problem)의 원인이 되는 고유의 모호성(inherent ambiguity), 폐쇄(occlusion), 광도 왜곡(photometric distortion), 형상 왜곡(figural distortion) 등의 문제점을 갖는다. 이러한 문제점들로 인해 대응점 정합(matching) 과정에서 에러가 발생하고, 에러는 인접 화소의 깊이정보로 전파된다. 이러한 단점을 극복하기 위해 영상 내 화소의 유사도에 관한 평가함수뿐만 아니라 변이정보의 분포 통계에 근거한 확률모델을 이용한 기법<sup>[4,5]</sup>과 신경망(neural network) 이론을 도입하여 연산 중 출력되는 정보를 정규화하고 학습효과를 이용하여 변이정보를 판단하는 기법<sup>[6]</sup>이 연구되었다. 또한 영상의 단일영역에 국한하지 않고 영상의 주사선(scan-line) 전체에 대한 화소들의 대응관계를 동적계획법(dynamic

programming)을 이용하여 풀어나가는 기법<sup>[7,8]</sup>도 연구되고 있다. 최근에는 변이정보의 정확도를 다른 관점에서 접근하여, 획득된 깊이정보의 정확도 향상을 목적으로 필터링 기법을 이용하는 기법<sup>[9,10]</sup>들이 활발하게 연구되고 있다.

일반적으로 이들 연구는 대부분 수행시간이 너무 길어서 실시간 비디오 서비스에 사용하기에는 적합하지 않고, 실험에 사용된 영상에만 특화되어 있어 일반적인 자연영상을 대상으로 수행할 경우 성능이 현저히 떨어지는 것을 확인할 수 있다<sup>[11]</sup>. 따라서 실시간성을 확보하기 위해서는 하드웨어로 개발되어야 하는데, 지금까지 깊이정보를 생성하기 위한 하드웨어를 최적화하는 연구는 많지 않다.

본 논문은 실시간 비디오 시스템에 적용할 수 있는 변이정보 연산기 구현 및 이에 적합한 변이정보 및 깊이정보의 생성 기법에 목표를 두고 있다. 이를 위하여 먼저 스테레오 비전에 대하여 분석하고, 기존 기법들이 영상의 특성에 기인하여 발생하는 단점을 개선하여 텍스처(texture) 등의 분포 특성을 고려한 가변 크기 윈도우 정합 기법을 적용한다. 마지막으로 제안한 기법은 최적화된 하드웨어 구조로 구현한다.

본 논문은 다음과 같이 구성된다. 2장에서는 스테레오 정합을 위한 비용함수와 정합 윈도우에 대해서 설명하고, 3장에서는 정합 방법을 제안한다. 4장에서는 제안한 하드웨어의 구조와 구현결과를 보이고, 5장에서 결론을 맺는다.

## II. 스테레오 정합

스테레오 정합은 좌우의 두 영상 중에서 한 영상을 기준영상으로 정하고 다른 영상을 대상영상으로 정한 뒤 기준영상의 특정 화소 혹은 영역을 대상영상에서 찾는 과정이다. 그림 1에는 우측 영상에서

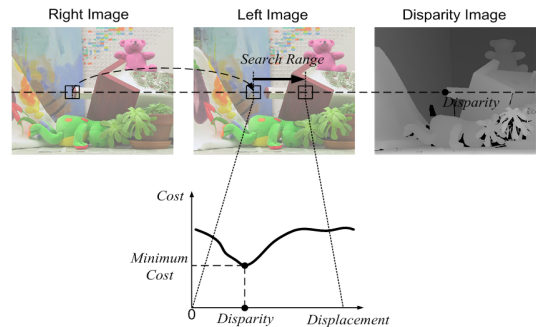


그림 1. 최소화 비용함수를 이용한 유사성 판단  
Fig. 1. Similarity measurement using minimize cost function

정합 윈도우로 정의된 특정 영역과 유사한 영역을 좌측 영상에서 탐색하는 것을 보이고 있다. 유사성을 찾기 위해서 최소화 비용함수를 이용한다. 탐색 구간에서 각각의 비용을 구한 뒤 최소의 비용을 갖는 변위를 변이벡터로 정의한다.

앞서 설명한 스테레오 정합 기법을 소프트웨어로 구현하여 각 비용함수 및 정합 윈도우의 크기가 갖는 특성을 분석하였다. 실험에 사용된 영상은 미국 middlebury 대학에서 제공하는 것들이다<sup>[16]</sup>.

### 2.1 비용함수

두 영상에서 대응점을 찾는 과정은 한 영상에서 특정 화소를 기준으로 인접 화소들의 집합을 의미하는 정합 윈도우와 유사성이 높은 윈도우를 다른 영상에서 찾는 것이다. 이러한 기법을 블록 정합 알고리즘(Block Matching Algorithm, BMA)이라 하고, MPEG<sup>[12]</sup> 및 H.264/AVC와 같은 동영상 압축 표준에서 움직임 예측할 때 사용한다. 유사성에 대한 정량적인 판단기준은 일반적으로 비용함수로 정의한다. 비용함수는 비용값에 대한 선택기준이 최소인지 최대인지에 따라서 최소화 비용 함수(SAD, SSD, MAD)와 최대화 비용 함수(NCC, MNCC)로 구분된다<sup>[15]</sup>.

비용함수는 특성과 변이정보에 대한 효과가 상이하다. 일반적으로 상관도 기반의 비용함수가 높은 정확도를 갖지만 수행시간이 길다는 단점으로 인하

여 선택적으로 사용된다. 표 1은 비용함수의 에 따라 수행시간을 측정한 결과를 나타내고 있다. 전체적으로 살펴보면 SAD 비용함수에 의한 연산이 가장 짧은 수행시간을 갖는다.

표 2는 비용함수에 따른 변이정보의 에러율이다. 변이정보의 에러율은 추출된 변이영상과 원래의 영상을 비교할 때 일치하지 않는 화소의 개수이다.

### 2.2 정합 윈도우의 효과

그림 2는 3×3 크기와 5×5 크기의 정합 윈도우를 사용하여 실험영상의 190번째 주사선에 대한 비용을 추적한 그림이다. 흰 선으로 표시된 부분은 최소 비용을 추적하여 표시한 것이고, 높이는 각 위치에서의 변이정보에 해당한다.

여기서 A영역과 B영역은 각기 다른 영상의 특성으로 구분된 것인데, A영역은 텍스처가 균일하여 비용함수의 변화가 적기 때문에 작은 크기의 정합 윈도우를 사용할 경우 변이정보가 제대로 추출되지 않음을 보이고 있다. B영역은 물체의 윤곽부분에 대한 특성을 설명하고자 나타낸 영역으로, B영역 경계는 영상 내 물체의 경계부분이며 배경과의 텍스처 변화가 심한 영역이다. 이러한 영역에서는 작은 크기의 정합 윈도우를 사용하는 경우, 비용값이 명확하게 변화하여 비용값에 대한 분포가 크다. 하지만 큰 크기의 정합 윈도우를 사용하는 경우 경계면에 위치한 비용값이 왜곡되어 부정확한 변이정보

표 1. 비용함수에 따른 수행시간의 비교  
Table 1. The comparison according to the cost function of the processing time (sec)

Cost Function	SAD	SSD	MAD	NCC	MNCC
Teddy	3.24	4.51	5.10	8.54	7.84
Cones	4.51	5.32	5.52	10.52	8.38
Tsukuba	3.62	5.16	5.32	9.48	7.81
Venus	3.87	4.98	5.52	9.84	8.29

표 2. 비용함수에 따른 변이정보 에러율의 비교  
Table 2. The comparison according to the cost function of the disparity error ratio (%)

Cost Function	SAD	SSD	MAD	NCC	MNCC
Teddy	30.8	30.8	31.5	20.2	21.3
Cones	33.5	33.5	32.2	22.2	25.6
Tsukuba	28.5	28.5	27.5	25.5	27.8
Venus	36.4	36.4	35.8	30.5	28.2

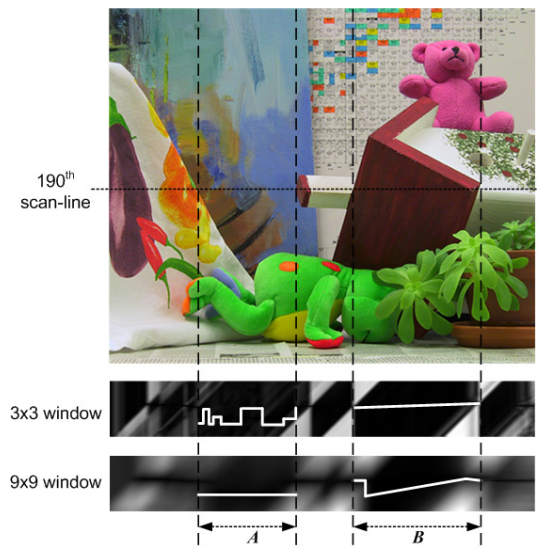


그림 2. 정합 윈도우의 크기 변화에 대한 변이 공간의 예제  
Fig. 2. An example of disparity space about a size of matching window

를 추출하게 된다.

### 2.3 변이 기울기 제약

변이 기울기 제한(disparity gradient limit) 제약은 영상 내에서 동일한 물체의 변이는 완만하게 변화해야 한다는 조건이고, 변이 평활(disparity smoothness) 제약 조건이라고도 한다. 공간상의 점 A와 점 B가 서로 인접한 위치에 존재한다고 가정하면, 점 A는 좌측 영상에  $A_L$ , 우측 영상에  $A_R$ 로 투영되고 점 B도 동일하게 각각  $B_L$ 과  $B_R$ 로 투영된다. 만약  $A_L$ 과  $A_R$ 이 서로 대응점으로 판별되었다면, 그 크기는 식 (1)과 같이 표현된다.

$$|(A_L - A_R) - (B_L - B_R)| \quad (1)$$

이 제약조건은 일반적으로 변이정보가 얻어진 후 변이영상을 기준으로 각 화소들간의 관계를 규정하여 변이영상을 개선하기 위한 기법으로 응용된다. 식 (1)의 크기에 임계값을 정하여 그 크기를 제한하면 변이를 조절할 수 있다. 변이영상을 후처리하는 과정에서 한 변이가 주변의 변이보다 상대적으로 너무 크거나 작은 값을 갖는 경우를 판별하여 보정할 수 있다. 그러나 물체의 경계에서 실제 변이를 왜곡할 수도 있기 때문에 선택적으로 적용하여야 한다.

## III. 영상 특성에 따른 가변 크기 정합

변이정보의 정확도가 감소하는 주요 요인은 영상 내 텍스처가 균일한 영역에서 변이정보가 정확하게 획득되지 않는다는 문제점과 객체의 경계 부근에서 정합 윈도우의 크기가 적합하지 못하여 변이정보가 왜곡되는 현상 때문이다. 본 장에서는 앞 장에서 설명한 비용함수와 정합 윈도우 크기를 바탕으로 변이정보의 정확도를 향상하기 위한 기법을 제안한다.

### 3.1 비용값의 분포에 대한 분석

텍스처의 복잡도에 따라 변이 정보의 정확도가 결정된다. 만약 영상의 복잡도가 낮다면 탐색구간 내 비용값에 대한 분포는 작다. 이러한 상태로 비용함수를 결정할 경우에 정합 윈도우 내의 잡음이나 인접 윈도우들과의 중복성으로 인하여 적절하지 않은 변이 벡터가 추출될 수 있다. 따라서 비용값을 선택하기 위한 탐색구간을 제한할 경우 정확한 비용값이 선택될 가능성이 높아진다. 그러나 전체 탐

색구간 중에서 탐색구간의 위치와 크기의 설정을 제한해야 하는 문제가 발생한다. 이 문제를 해결하기 위한 과정은 다음과 같다.

- ① Step1 : 입력되는 두 영상에 대하여 비교적 큰 크기의 정합 윈도우를 사용하여 영상 내 물체의 대략적인 변이정보를 구한다.
- ② Step2 : step1에서 획득된 변이정보를 현재 화소 변이값의 기준으로 정하고 새로운 탐색구간을 정의한다.
- ③ Step3 : step2에서 얻어진 탐색구간을 바탕으로 작은 크기의 정합 윈도우를 사용하여 다시 비용연산을 수행한다.

그림 3에서는 이러한 과정을 보이고 있다. 탐색구간의 범위는 실험적으로 구하였는데 이에 대한 자세한 결과는 다음 절에서 소개하기로 한다.

이러한 과정을 거치면 변이정보는 큰 크기의 정합 윈도우를 사용한 경우에 비하여 텍스처가 적은 영역에 대해서도 비교적 정확하게 추출된다. 이 기법에 적용된 결과를 그림 4에 나타내었다.

그림 4에서 결과에 대한 어려움은 그림 5(a), (b), 및 (c)가 각각 64.3%, 27.%, 및 31.6%이다. 정량적인 결과만을 놓고 보면 그림 4(b)의 결과가 좋음을 알 수 있다. 하지만 앞 장에서 설명한 변이정보의 특성을 고려하면 물체의 윤곽에 대한 시각적인 정보는 오히려 그림 4(c)가 우수함을 알 수 있다. 그

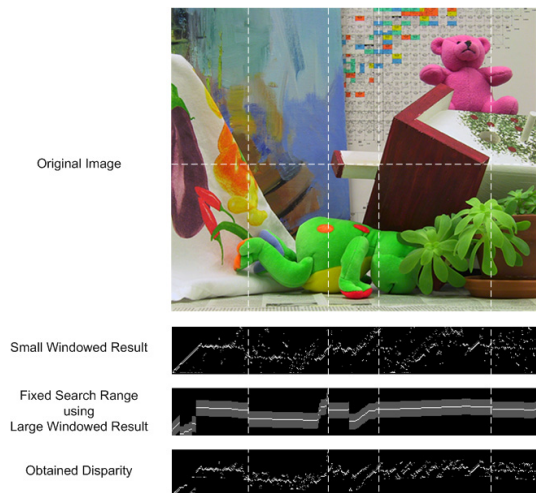


그림 3. 하이브리드 워터마크 추출을 위한 광학시스템  
Fig. 3. Optical system for hybrid watermark extraction

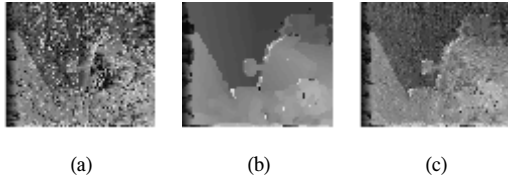


그림 4. 고정영역 탐색 기법이 적용된 변이정보 추출 결과; (a) 작은 크기의 정합 윈도우, (b) 큰 크기의 정합 윈도우, (c) 고정영역 탐색 기법

Fig. 4. The result of a disparity extraction which a fixed search range technique is applied; (a) disparity using small size matching window, (b) disparity using large size matching window, (c) disparity which the fixed search range technique is applied

림 4(b)의 에러율이 낮은 이유는 영상의 많은 영역에서 텍스처가 균일하기 때문이다. 그림 4(c)는 영상의 전 영역에 대해 일정한 탐색구간이 정의되었기 때문에 성능이 낮다. 따라서 정확도를 더욱 향상시키기 위해서는 텍스처의 복잡도에 따라 탐색구간의 크기가 선택적으로 정의되어야 한다.

### 3.2 텍스처의 복잡도에 대한 분석

영상의 복잡도는 영상의 변화율을 구함으로써 얻을 수 있다. 윈도우 내에서 화소간의 미분값을 구한 후에 그 값의 절대치를 누적함으로써 영상의 복잡도를 측정한다. 현재 정합하고자 하는 블록에 대해서 식 (2)와 같은 국부 변화도(Local Variation, LV) 함수를 이용하여 영상의 복잡도를 구한다.

$$LV(u, v, d) = \sum_{(i,j)} |I_L(u+i, v+j) - \mu| \quad (2)$$

여기서  $\mu$ 는 해당 윈도우의 평균값이다. 영상 내 복잡한 영역에서는 그 LV가 높고 그렇지 않은 영역에서는 에너지가 낮다.

영상의 복잡도를 나타내는 LV의 분산은 매우 크기 때문에 이 값을 그대로 새로운 탐색구간으로 이용하기에는 적합하지 않다. 원 영상의 국부 변화량을 구한 뒤 그 크기에 따라 새로운 탐색구간을 찾고 최소 비용값을 탐색한다. 그림 5는 이러한 과정을 나타낸 것이다.

### 3.3 가변 크기 정합 윈도우

앞 절에서 탐색영역을 제한하거나 텍스처의 복잡도에 따라 탐색구간을 변화시키는 기법이 유효함을 확인하였다. 하지만 이러한 기법은 먼저 큰 크기의 정합 윈도우를 사용하여 대략적으로 변이정보 획득되어야 한다는 단점이 있다. 이는 수행시간을 불필

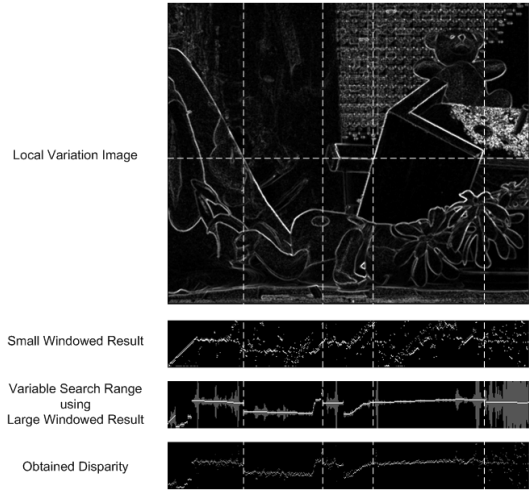


그림 5. 가변영역 탐색 기법의 예  
Fig. 5. An example of variable search range technique

요하게 증가시키는 요인이기 개선된 방법이 요구된다.

정합 윈도우의 크기에 따라 비용값의 분포가 달라진다. 이때 중요한 점은 분포의 정도에 따라 변이정보의 정확도가 결정된다는 것이다. 비용값의 분포가 클수록 높은 정확도의 변이정보를 얻을 수 있다. 따라서 비용값의 분포 범위를 증가시키기 위해 정합 윈도우의 크기를 고정하지 않고 정합 윈도우의 크기를 특정 조건(LV)에 따라서 변화시킨다. 이 순서를 정의하면 다음과 같고, 이 과정을 그림 7에서 나타내었다.

- ① Step1 : 현재 정합하고자 하는 윈도우에서 고정 크기로 LV를 구한다.
- ② Step2 : LV의 크기에 임계치를 부여하여 정합 윈도우의 크기를 결정한다.
- ③ Step3 : step2에서 얻어진 정합 윈도우의 크기로 최소 비용값 탐색을 수행한다.

그림 6은 수정된 기법을 이용하여 변이정보를 얻은 결과이다. 그림 4(c)의 에러율은 31.6%였지만, 수정된 기법을 적용한 결과 그림 6(c)의 에러율은 28.4%로 감소하였다.

본 논문에서는 LV의 크기에 따라 정합 윈도우의 크기를 3×3, 5×5, 그리고 7×7 크기의 세 가지로 구분한다. 실험적으로 정합 윈도우의 크기가 그 이상이거나 이하일 경우에 결과가 향상되지 않는다. 정합 윈도우의 크기를 결정하는 임계값은 실험적으로 구하였다. 그림 8은 제안한 기법을 적용하여 변

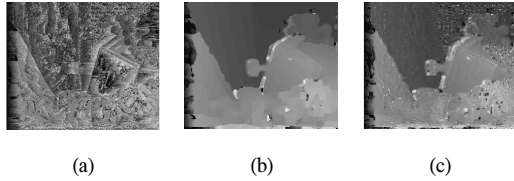


그림 6. 가변영역 탐색 기법이 적용된 변이정보 추출 결과; (a) 작은 크기의 정합 윈도우가 적용된 결과, (b) 큰 크기의 정합 윈도우가 적용된 결과, (c) 가변영역 탐색 기법이 적용된 결과

Fig. 6. The result of a disparity extraction which a variable search range technique is applied; (a) disparity using small size matching window, (b) disparity using large size matching window, (c) disparity which the variable search range technique applied

이정보를 추출한 결과이다. 그림 8(c)의 에러율은 23.7%로 앞서서 설명한 기법 중 가장 우수하다. 그림을 살펴보면 물체의 윤곽 부분에서도 비교적 정확한 변이정보가 추출되었음을 시각적으로 확인할 수 있다.

연산 과정에서 동시성을 고려한다면 LV 연산은

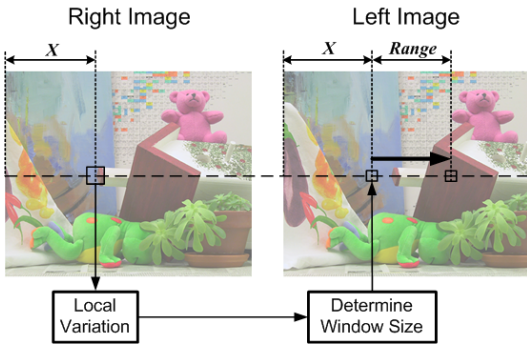


그림 7. 가변 크기 정합 윈도우의 예제  
Fig. 7. An example of variable size matching window technique

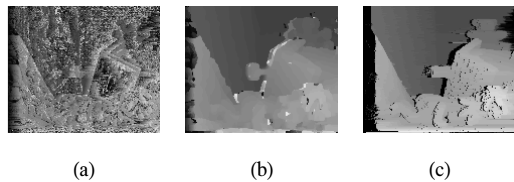


그림 8. 가변 크기 정합 윈도우 기법이 적용된 변이정보 추출 결과; (a) 작은 크기의 정합 윈도우가 적용된 결과, (b) 큰 크기의 정합 윈도우가 적용된 결과, (c) 제안한 기법이 적용된 결과

Fig. 8. The result of a disparity extraction which a variable size matching window technique is applied; (a) disparity using small size matching window, (b) disparity using large size matching window, (c) disparity which the proposed technique applied

탐색구간에서 비용값을 연산하는 과정과 동시에 수행이 가능하다. 만약 세 가지 정합 윈도우에 대한 변이정보를 동시에 연산하고 이 연산과정 내에서 추출되는 LV에 따라 각 정합 윈도우에 얻어진 변이정보중 하나를 최종 변이정보로 선택한다면 LV에 따라 정합 윈도우의 크기를 결정하는 수행시간을 생략할 수 있다. 다음 장에서는 이러한 기법이 적용된 하드웨어에 대해서 설명할 것이다.

#### IV. 변이추출 프로세서

본 장에서는 제안한 변이정보 추출 알고리즘을 하드웨어로 설계한다. 변이정보를 추출하기 위해서는 많은 연산량이 필요하기 때문에 변이정보 연산기의 핵심 블록에 해당하는 비용 연산기를 효율적으로 설계해야만 한다. 변이정보의 정확도를 향상시키기 위해서 가변크기의 정합 윈도우 구조를 적용한다. 특정 화소에 대한 탐색구간은 인접한 화소들에 대한 탐색구간과 중복된다. 이러한 중복성을 최대한 활용하면 비용 연산기의 입력을 다시 버퍼링할 필요가 없고, 수행시간을 단축시킬 수 있다.

그림 9에서는 최상위 블록에 해당하는 변이추출 프로세서의 하드웨어 구조를 나타내었다. 변이추출 프로세서는 왼쪽 및 오른쪽 영상을 외부로부터 입력받고 처리하는 *Input Interface*, 입력되는 영상을 외부의 *SRDAM*에 저장하고 변이를 추출하는데 필요한 영상 정보를 *SDRAM*으로부터 읽어들이는 *Memory Controller*, 기능 블록들을 제어하는 *Chip Controller*, 추출된 변이정보를 외부로 출력하는 *Output Interface*, 그리고 핵심블록인 변이정보를 추출하는 *Disparity Estimator*로 구성된다. 본 논문에서는 주로 *Disparity Estimator*에 대해서 설명하고자 한다.

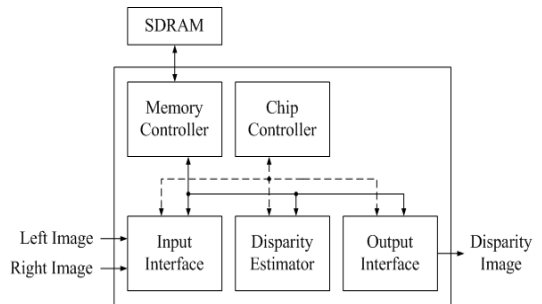


그림 9. 변이 추출 프로세서의 구조  
Fig. 9. Architecture of disparity estimation processor

### 4.1. 변이 추출기 (Disparity Estimator)

외부 메모리에 저장되어 있던 영상은 *Memory Controller*를 통해 호출된 후에 윈도우 크기에 따라 버퍼링되고, 버퍼링된 화소값들은 비용 연산기로 입력된다. 비용 연산기는 블록 정합기법을 이용하여 최소비용을 찾기 위한 연산을 수행한다. 하드웨어의 장점 중 하나인 병렬성과 동시성을 도입하여 세 개의 정합 윈도우에서 추출된 변이정보들이 동시에 출력되도록 구성한다. 텍스처가 적은 영역에 대하여 정확한 변이정보 추출을 위하여 작은 크기의 정합 윈도우와 중간 크기의 정합 윈도우, 큰 크기의 정합 윈도우로 동시에 연산하여 비용 연산 결과인 세 종류의 변이정보가 출력된다. 이와 동시에 현재 정합하고 있는 윈도우의 국부 변화량 연산을 수행하고, 그 결과에 따라 최종 변이정보를 선택한다. 또한 이미 연산과정이 끝난 인접 화소의 변이정보를 참조하여 변이 기울기 제한조건에 따라 화소간 변이정보가 너무 크게 변하지 않도록 조절한다.

그림 10은 본 논문에서 설계한 변이정보 연산기의 전체 구조이다. *Right/Left Window REG*는 외부 저장매체로부터 입력된 윈도우 및 탐색영역의 화소들을 버퍼링한다. *MCSM*은 비용 연산기에서는 각각 크기가 다른 정합윈도우의 비용연산에 따른 각각의 변이정보를 출력한다. *Local Variation Calculation*은 현재 정합윈도우에 대한 국부 변화량을 연산한다. *Final Disparity Selector*는 얻어진 국부 변화량으로 세 종류의 변이정보 중 최종 변이정보를 선택하고, 미리 추출된 인접화소들의 변이정보를 참조하여 최종 변이정보를 출력한다.

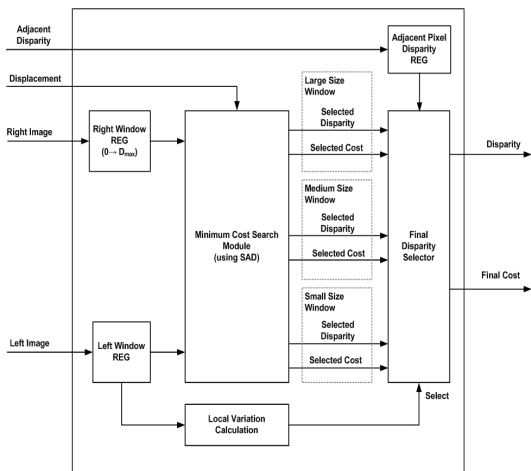


그림 10. 변이 추출기의 구조  
Fig. 10. Architecture of disparity estimator

### 4.2. 세부 모듈의 구현

#### 4.2.1 정합 윈도우 버퍼

한 주사선에서  $x$ 번째 화소의 변이정보를 추출하기 위한 탐색구간과  $x+1$ 번째 화소의 변이정보 추출을 위한 탐색구간은 거의 동일하다. 즉, 탐색구간의 화소들은 공간적 중복성을 갖는다. 저장된 데이터에 접근하는데 여러 클럭이 소요되는 SDRAM으로부터 중복되는 화소들을 연산할때마다 호출한다면 너무 많은 시간이 소요된다. 따라서 전체적인 수행시간이 증가하고, 하드웨어의 성능이 감소된다. 이러한 비효율성을 최소화하기 위한 하드웨어 구조가 필요하다.

공간적 중복성을 최대도 활용하여 외부 SDRAM 접근을 최소화 한다.  $x$ 번째 화소의 변이정보 연산에 추출에 사용된 탐색구간의 화소들을  $x+1$ 번째 화소 변이정보 연산 시 다시 사용하는 것이다. 그림 11은 제안한 정합 윈도우 버퍼의 구조이다.

그림 11에 나타난 버퍼구조는  $3 \times 3$ 크기 정합 윈도우에 대한 것으로 최대 탐색영역을  $D\_MAX$ 로 나타내었다. 각 블록은 화소값이 저장되는 레지스터인데 회색으로 음영 처리된 레지스터는 비용 연산기의 입력이 된다.

외부 SDRAM에 저장된 화소값을 순차적으로 읽어서 횡방향 라인버퍼에 저장한다. 이러한 라인버퍼링을 탐색구간만큼 수행하고, 탐색구간의 버퍼링이

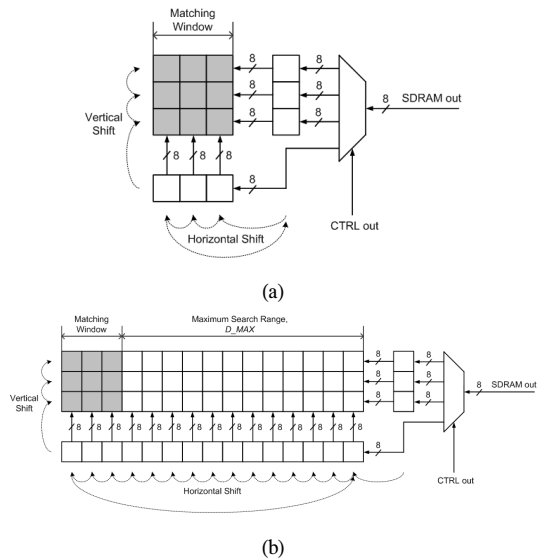


그림 11. 정합 윈도우 버퍼의 최종 구조; (a) 좌영상에 대한 정합 윈도우 버퍼, (b) 우영상에 대한 탐색구간 윈도우 버퍼  
Fig. 11. The structure of matching window buffer; (a) matching window for left image, (b) search range window buffer for right image

끝나면 수직방향으로 쉬프트하여 탐색영역의 한 스캔라인에 대해서 저장을 완료한다. 이 과정을 다음 2개의 스캔라인에 대하여 반복하면 현재 탐색영역에 대한 버퍼링이 완료된다. 이러한 구조는 정합 대상 윈도우를 매번 버퍼링할 필요가 없으므로 SDRAM 접근에 따른 지연시간을 피하고 단순 시프트 동작으로서 수행이 가능하다는 장점이 있다. 비용 연산 과정은  $D\_MAX$  사이클 동안 이루어지는데 좌측으로 쉬프트하고 가장 좌측 레지스터의 값은 다시 가장 우측으로 저장되는 배럴 쉬프트(barrel shift)구조를 사용하였다. 이것은 한 화소에 대한 탐색영역은 다음 화소에 대한 탐색영역과 1화소가 밀려난 형태이기 때문이다. 다음 화소에 대한 변이정보를 추출할 시에는 우측으로 새로운  $1 \times 3$  화소의 탐색영역이 필요하다. 이에 대한 버퍼링은 비용 연산 과정과 동시에 수행될 수 있으므로 동작시간의 증가를 야기하지 않는다.

#### 4.2.2 비용 연산기

정합 윈도우의 유사도 비교를 위하여 SAD 기반의 BMA를 사용한다. 연산결과를 비교하여 최소의 비용값에 따라 변이정보를 할당하는 것이다. SAD 연산은 차이값을 지속적으로 누적하는 연산이므로 하드웨어 구조는 덧셈기, 레지스터 및 feedback 루프를 간단히 구성할 수 있다.

변이추출 연산은 동영상 압축 기법에서 화면 간 예측구조에 사용되고 있는 움직임 예측과 다르다. 움직임 예측은 블록 단위로 정합하고 최적의 비용에 따라 블록 단위의 움직임벡터를 출력하지만 변이 연산은 은 영상 내 모든 화소에 대하여 변이벡터를 추출한다. 이와 같이 변이추출은 동영상 압축 기술의 움직임 예측구조에 비하여 연산량이 매우 크기 때문에 효율적인 iterative computation 구조에 따라 변이추출 하드웨어의 성능이 좌우된다. 제안한 하드웨어는 성능을 최대를 증가시키기 위해서 세 개의 윈도우에 대한 연산을 병렬로 수행할 수 있도록 하고, 비용함수를 연산하는 블록은 100% 파이프라인 구조를 갖도록 한다. 그림 12는 MCSM의 구조를 나타낸다.

그림 13은 그림 12의  $3 \times 3$  Cost Calculator에 해당한다. 최소비용의 선택은 모든 가능한 비용연산이 끝난 후 이루어지지만 순차적인 덧셈연산이 이루어지는 동시에도 가능하다. 그림 13에서 최초 정합 윈도우의 비용연산 결과는 6사이클 후 출력되며 이후 매 사이클마다 다음 정합 윈도우의 비용연산 결

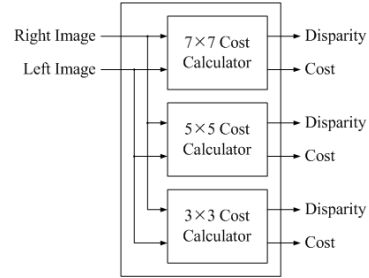


그림 12. MCSM의 구조  
Fig. 12. Architecture of MCSM

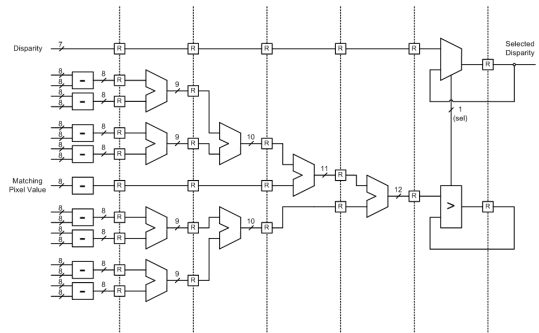


그림 13.  $3 \times 3$  크기 비용 연산기의 게이트 수준 구조  
Fig. 13. Gate level structure of a  $3 \times 3$  size cost computation module

과가 출력된다. 그림 13에서 볼 수 있듯이 비용누적 과정과 최소값 선택 과정은 병렬적으로 수행된다.

제안한 하드웨어는 앞서서 설명한 것과 같이 가변 크기 정합 윈도우 구조를 사용하였다. 이러한 연산은 기본적으로 각 크기의 정합 윈도우의 결과를 비교하여 현재 정합하고자 하는 윈도우의 복잡도에 따라 세 종류의 변이정보 중 하나의 변이정보를 최종 변이정보로 선택하게 된다. 즉, 크기가 다른 각각의 정합 윈도우에 대한 연산기가 필요하다. 그림 13의  $3 \times 3$  크기의 연산기를 기본으로  $5 \times 5$  크기와  $7 \times 7$  크기로 확장하여 구현할 수 있다.

#### 4.3 하드웨어 구현 결과

본 논문에서 설계한 하드웨어는 Verilog-HDL을 이용하여 설계하였다. 설계는 HDL 상향식 설계 기법을 통해서 이루어졌으며 특정 구현 대상에 국한하지 않는 범용적인 설계를 이루고자 오직 IEEE 표준 라이브러리만을 사용하였다. 설계한 H/W를 검증하기 위해 Synopsys社의 Design Compiler<sup>TM</sup>로 Hynix 0.35um CMOS cell library를 사용하여 논리합성(logic synthesis)을 수행하였다. 합성 결과를 토대로 Cadence社의 NC-Verilog<sup>TM</sup>을 사용하여 시물



레이션을 수행한 결과 최대 120MHz에서 동작하고 안정적으로 100MHz에서 동작함을 확인하였다.

표 3에서는 본 논문에서 구현한 하드웨어와 기존 하드웨어의 성능을 비교하였다. 동작 주파수를 고려해보면 Niitsuma<sup>[18]</sup>의 연구가 프레임율은 앞서나 논 본문에서는 가변 크기 윈도우를 도입하였고 비용값 탐색구간의 크기도 월등하기 때문에 변이정보의 정확도는 더 우수하다.

본 하드웨어가 가지는 임계 경로(critical path)는 버퍼링 된 정합 윈도우 내의 화소값에 대하여 차이 (difference)연산 결과가 음수가 나오지 않도록 큰 값과 작은 값으로 정렬하는 단계에서 발생한다. 따라서 이 연산단계를 개선하여 향상된 효율적인 구조를 치환하면 더 좋은 성능을 나타낼 수 있을 것으로 기대된다.

표 3. 하드웨어 구현 결과  
Table 3. Implementation results

Feature	Our work	Niitsuma <sup>[18]</sup>
Image size	VGA (640×480)	VGA (640×480)
Window size	Variable size (3×3, 5×5, 7×7)	Fixed Size (7×7)
Cost function	SAD	SAD
Max. disparity	60	27
Frame Rate	15 fps	30 fps
Platform (Gate count/Area)	ASIC (235K gates)	FPGA (N/A)
Technology	Hynix 0.35um CMOS cell	Xilinx XC2V6000 <sup>TM</sup>
Voltage	+3.2V, -0.0V	+1.8V, -0.0V
Max. frequency	120Hz	68MHz

## V. 결 론

본 논문에서는 실시간 다시점 비디오 서비스에서 가장 시점에서의 영상 렌더링에 사용되는 변이정보를 추출하기 위한 효율적인 기법을 제안하였고 이를 최적화된 하드웨어 구조로 구현하였다.

변이정보의 추출기법은 이미 많은 연구사례가 있으나 이들 연구는 수행시간이 너무 길어 본 연구에서 구현하고자 하는 실시간 비디오 서비스에 사용하기에는 적합하지 않다. 따라서 실시간성을 확보하기 위해서는 하드웨어 구현에 적합한 효율적인 기법이 필요하며, 본 논문에서는 이를 위하여 고속 연산이 가능한 화소 대 화소의 움직임 추정 기법을 이용하였다. 이러한 움직임 추정 기법이 가지는 단점은 대응점을 정합 윈도우간의 유사성에만 의존하

여 텍스처의 복잡도가 적은 영역에서는 정확도가 떨어지는 단점이 있다는 것인데 이를 방지하고자, 정합 윈도우의 복잡도에 따라 정합 윈도우의 크기를 가변시켜 비용값이 왜곡되는 현상을 방지함으로써 변이정보의 정확도를 향상하였다.

하드웨어 구조에서는 본 논문에서 제안한 기법이 가지는 각 연산 단계의 시간적 중복성을 최대한 활용함으로써 고속 연산이 가능하도록 하였다. 이는 핵심 모듈인 비용 연산기를 여러 크기의 정합 윈도우에 대한 비용이 병렬적으로 연산되도록 설계하였다. 하드웨어의 장점을 최대한 활용하여 일정 싸이클 내에 각 정합 윈도우 블록이 동시에 출력되는 파이프라인 구조를 채택하였다. 또한 정합 윈도우 내의 화소가 저장되는 버퍼는 인접한 화소간 비용값의 탐색구간이 가지는 공간적 중복성을 최대한 활용하여 외부 저장매체로부터의 읽기 동작에 따른 대기시간의 발생을 최소화하였다.

본 논문에서 구현한 하드웨어는 효율적인 자원사용 및 병렬연산 구조를 가진다. 따라서 비교적 간단한 확장을 통하여 더 높은 해상도의 변이정보 추출 연산에도 적용 가능하다. 또한 이러한 대응점 선별에 관한 문제는 동영상 압축 기법의 움직임 추정 기법에도 이용된다. 따라서 비용 연산기의 경우 움직임 예측 하드웨어에서도 그대로 적용할 수 있으며 전체 하드웨어 설계 시 높은 효율을 가질 수 있을 것으로 사료된다.

## 참 고 문 헌

- [1] ISO/IEC MPEG & ITU-T VCEG, "Multiview video plus depth (MVD) format for advanced 3D video systems," JVT-W100, April 2007.
- [2] ISO/IEC, "ISO/IEC JTC1/SC29/WG11 Coding of Moving Picture and Audio," Draft of version 4 of ISO/IEC 14496-10 (E) MPEG05/N7081, April 2005.
- [3] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," International Journal of Computer Vision, Vol. 47, Issue 1-3, pp. 7-42, April 2002.
- [4] J. Sun, N. N. Zheng, and H. Y. Shum, "Stereo matching using belief propagation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, Issue 25, pp. 787-800,

July 2003.

[5] P. N. Belhumeur, "A Bayesian approach to binocular stereopsis," *International Journal of Computer Vision*, Vol. 19, Issue 3, pp. 237-260, Aug. 1996.

[6] I. Gallo, E. Binaghi, and M. Raspanti, "Neural disparity computation for dense two-frame stereo correspondence," *Pattern Recognition Letters*, Vol. 29, Issue 5, pp. 673-687, April 2008.

[7] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, March 1985.

[8] C. J. Tsai and A. K. Katsaggelos, "Dense disparity estimation with a divide-and-conquer disparity space image technique," *IEEE Transactions on Multimedia*, Vol. 1, Issue 1, pp. 18-29, March 1999.

[9] C. Georgoulas, L. Kotoulas, G. Ch. Sirakoulis, I. Andreadis, and A. Gasteratos, "Real-time disparity map computation module," *Microprocessors and Microsystems*, Vol. 32, Issue 3, pp. 159-170, May 2008.

[10] D. I. Han, B. M. Lee, J. I. Cho, and D. H. Hwang, "Real-time object segmentation using disparity map of stereo matching," *Applied Mathematics and Computation*, Vol. 205, Issue 2, pp. 770-777, Nov. 2008.

[11] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, Issue 8, pp. 993-1008 Aug. 2003.

[12] ISO/IEC 11172-2, "Information technology coding of moving picture and associated audio for digital storage media at up to about 1.5Mbps," *International Standard*, 1993.

[13] ISO/IEC 11172-2, "Information technology coding of moving picture and associated : video," *International Standard*, 1995.

[14] ISO/IEC 14496-2, "Information technology coding of audio-visual object," *International Standard*, 2001.

[15] R. C. Gonzales and R. E. Woods, "Digital image processing," Prentice Hall, 2nd edition, 2001.

[16] <http://vision.middlebury.edu/stereo>

[17] Y. Wang, J. Ostermann, and Y. Q. Zhang, "Video Processing and Communications," Prentice hall, 2002.

[18] H. Niitsuma, and T. Maruyama, "Real-time detection of moving object," *Lecture Notes in Computer Science*, Vol. 3203, pp.1155-1157, 2004.

신 완 수 (Wan-Soo Shin)

정회원



2005년 2월 광운대학교 전자재료공학과(공학사)  
 2009년 2월 광운대학교 대학원(공학석사)  
 <관심분야> Multi-view video coding, Signal and image processing, FPGA/ASIC 설계

최 현 준 (Hyun-Jun Choi)

정회원



2003년 2월 광운대학교 전자재료공학과(공학사)  
 2005년 2월 광운대학교 대학원 졸업(공학석사)  
 2009년 2월: 광운대학교 대학원 졸업(공학박사)  
 2009년 3월~2009년 8월: 광운대학교 실감미디어 연구소 박사후 연구원  
 2009년 9월~현재 광운대학교 실감미디어 연구소 연구교수  
 <관심분야> Hybrid Image Processing, Contents Security, FPGA/ASIC 설계

서 영 호 (Young-Ho Seo)

중신회원



1999년 2월 광운대학교 전자재료공학과(공학사)  
2001년 2월 광운대학교 대학원 졸업(공학석사)  
2000년 3월~2001년 12월 인티스닷컴(주) 연구원  
2003년 6월~2004년 6월 한국전기연구원 연구원

2004년 8월 광운대학교 대학원졸업(공학박사)  
2004년 9월~2004년 11월 유한대학 겸임교수  
2004년 12월~2005년 8월 유한대학 연구교수  
2005년 9월~2008년 2월 한성대학교 정보통신공학과 교수  
2008년 3월~현재 광운대학교 교양학부 교수  
<관심분야> Image Processing/Compression, 워터마킹, 암호학, FPGA/ASIC 설계

김 동 욱 (Dong-Wook Kim)

중신회원



1983년 2월 한양대학교 전자공학과 졸업(공학사)  
1985년 2월 한양대학교 대학원 졸업(공학석사)  
1991년 9월 Georgia공과대학 전기공학과 졸업(공학박사)  
1992년 3월~현재 광운대학교 전자재료공학과 정교수

2009년 3월~현재 광운대학교 실감미디어연구소 연구소장  
2006년 3월~(사)실감미디어산업협회 이사  
<관심분야> 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication