컴퓨터 생성 홀로그램의 하드웨어 구현을 위한 버스 구조 분석

서영호* · 김동욱*

Bus Architecture Analysis for Hardware Implementation of Computer Generated Hologram

Yong-Ho Seo* · Dong-Wook Kim*

이 논문은 2012년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

요 약

최근 차세대 영상 기술로 홀로그래피가 많은 주목을 받고 있다. 홀로그램은 광학적인 촬영을 통해서 획득할 수도 있지만 최근에는 컴퓨터를 이용한 홀로그램 생성 방법을 많이 사용하고 있다. 이를 컴퓨터 생성 홀로그램 (computer generated hologram, CGH)이라 하는데 CGH는 많은 연산량이 요구되어 S/W를 이용하면 실시간으로 생성하는 것이 불가능하다. 따라서 실시간의 CGH를 위해서 FPGA나 GPU를 이용한 연산 방법이 주로 사용되고 있다. 하드웨어를 기반으로 하여 구현할 경우에 내부 시스템의 비트 제한으로 인하여 S/W와 같은 품질을 얻을 수 없다. 따라서 본 논문에서는 품질의 저하를 최소화하면서 하드웨어의 자원을 최대한 감소시킬 수 있는 하드웨어 비트 너비를 분석하여 가이드라인을 제시하고자 한다. 이를 위해서 1비트 단위의 고정소수점 시뮬레이션을 모든 내부 변수 및 연산과정에 대해 수행하고, 수치적인 결과와 시각적인 결과를 종합적으로 분석하여 최적의 비트 너비와 응용 분야에 따른 비트 너비를 제시한다.

ABSTRACT

Recently, holography has received much attention as the next generation visual technology. Hologram is obtained by the optical capturing, but in recent years it is mainly produced by the method using computer. This method is named by computer generated hologram (CGH). Since CGH requires huge computational amount, if it is implemented by S/W it can't work in real time. Therefore it should use FPGA or GPU for real time operation. If it is implemented in the type of H/W, it can't obtain the same quality as S/W due to the bit limitation of the internal system. In this paper, we analyze the bit width for minimizing the degradation of the hologram and reducing more hardware resources and propose guidelines for H/W implementation of CGH. To do this, we performs fixed-points simulations according to main internal variables and arithmetics, analyze the numerical and visual results, and present the optimal bit width according to application fields.

키워드

컴퓨터 생성 홀로그램, 하드웨어, FPGA, GPU, 고정소수점 시뮬레이션

Key word

Computer generated hologram, hardware, FPGA, GPU, fixed-point simulation

* 종신회원 : 광운대학교(김동욱, dwkim@kw.ac.kr)

접수일자 : 2011. 11. 15

심사완료일자 : 2011. 12. 09

Open Access <http://dx.doi.org/10.6109/jkiice.2012.16.4.713>

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서 론

홀로그래피는 1948년 Gabor에 의해 최초로 제안된 이래 3차원 정보를 기록할 수 있다는 특징 때문에 많은 연구자들의 관심을 끌어들였다. 기존의 홀로그래피는 홀로그램 필름에 3차원 정보를 기록하고, 현상된 필름을 사용하여 3차원 물체를 복원하는 방식을 사용함으로써 그 응용이 크게 제한되었다. 이러한 단점을 극복하기 위한 새로운 접근방법으로 1966년 이후 많은 연구자들이 컴퓨터에 의한 홀로그램(computer-generated hologram, CGH)의 제작을 연구해 오고 있다[1]. 이 기술은 물체파(object wave)와 기준파(reference wave)의 간섭에 의해 생성되는 간섭향을 계산함으로써 현실에서는 불가능한 이상적인 특성을 가진 부품을 제작하거나 특성시험 등을 위해 개발되었다[2][3].

CGH를 이용하여 한 프레임에 해당하는 홀로그램을 생성하기 위해서는 많은 연산량과 시간이 소요되기 때문에 고속의 연산방법이 필요하다. 이런 고속 CGH를 위한 여러 알고리즘들이 개발되어 왔다[4~6]. MIT Media lab의 Spatial Imaging Group(지금은 Object-based Media Group)은 고속 CGH를 위한 연구를 가장 먼저 시작한 연구그룹이다[4].

여기서 HPO(horizontal-parallax-only) 방식의 CGH 기법을 이용해 디지털 홀로그램을 생성하는 연구를 수행하였다. 이 연구에서는 LUT(Look-up Table)방식과 parallel supercomputer를 이용해 10,000 point의 light source 영상으로부터 1초에 한 장씩 디지털 홀로그램(해상도: 6M)을 생성하였다. 일본의 Nihon대학의 Yoshikawa교수는 root연산을 Taylor 전개를 통해 변형한 후 CGH 수식의 근사화를 통해 고속화 알고리즘을 정리하였다[5]. 일본의 Chiba 대학의 연구팀은 Yoshikawa교수의 이론을 변형시켜 x축에서 반복덧셈만을 수행하여 CGH를 계산하는 알고리즘을 제안하고 FPGA를 이용한 하드웨어로 구현하였다[6]. 이와 같이 CGH 연산량이 너무나 방대하기 때문에 실제로 소프트웨어로 CGH를 실시간으로 처리하는 것은 불가능하고 하드웨어로 구현되어야 하며, 지금까지 CGH를 위해 하드웨어를 활용한 많은 연구가 진행되어 왔다[7~14]. 이러한 연구들은 GPU 기반의 소프트웨어 방식[7~11]과 FPGA 기반의 하드웨어 칩 구현 방식[6][12~15]으로

나누어진다.

FPGA를 비롯하여 GPU도 내부적인 연산을 수행함에 있어서 비트 제약을 갖는다. 이러한 비트 제약은 이들을 이용하여 수행한 결과의 품질을 낮추거나 기대한 품질과 다른 결과를 가져오기도 한다. 따라서 내부적인 하드웨어의 비트 너비를 정확히 파악하고 그에 따른 알고리즘 수준에서의 비트 너비를 조절한다면 예측가능하면서 최소화된 품질의 결과를 얻을 수 있다. CGH 연산은 매우 큰 수들과 매우 높은 정밀도의 부동소수점이 함께 연산되는 경우가 많은 특수한 연산이 혼재되어 있어서 하드웨어 기반의 구현 시 이러한 알고리즘 수준에서 하드웨어 특성을 고려해야만 정확한 결과를 얻을 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 CGH 알고리즘에 대해서 설명한다. 3장에서는 고정소수점 실험 방법에 그에 따른 정량적인 결과를 보인다. 4장에서는 홀로그램을 복원한 결과들을 보이고 이에 따른 시각적인 분석 결과를 설명한다. 마지막으로 5장에서는 결론을 맺는다.

II. 컴퓨터 생성 홀로그램

디지털 홀로그램은 간섭무늬를 광학 필름 대신에 CCD 카메라에 기록하고 비디오 신호로 전송 혹은 저장한다. 기록된 홀로그램을 재생하기 위해서는 spatial light modulator(SLM)에 표시된 간섭무늬 데이터에 레이저광을 조사한다. 그림 1에서 디지털 홀로그램의 시스템 구성을 보이고 있다. 디지털 홀로그램은 기존의 광학 홀로그램에 의한 기법과 동일하게 레이저광을 집광 렌즈로 평행광을 만들고, 빔 분리기로 참조파와 객체파로 나눈다. 객체파는 피사체(object)에 조명된 다음 참조파와 직접 CCD에 조사되어 간섭무늬를 형성한다. 간섭무늬 정보는 SLM에 인가되고 여기에 평행광을 조사하면 1차 회절광이 발생하여 3차원 객체를 재생할 수 있다[2].

그림 1. 디지털 홀로그램 (a) 기록 (b) 복원
Fig. 1 Digital hologram (a) recoding (b) reconstruction

홀로그램은 그림 1과 같은 광학계를 이용하여도 취득할 수 있지만 이러한 광학계 자체를 수학적으로 모델링한 후에 연산에 의해서 취득할 수도 있다. 이러한 수학적 연산을 통해 얻어진 홀로그램을 컴퓨터 생성 홀로그램(computer-generated hologram, CGH)이라고 한다. 여러 종류의 CGH가 있지만 본 논문에서는 “Phase” 방식을 사용한다. 그림 1(a)에서 객체에서 CCD로 입사되는 파에서 위상 성분만을 이용하여 홀로그램을 생성하는 것이고 그 증명[3]은 본 논문에서 다루지 않는다.

파면의 간격이 시간과 공간에 걸쳐 일정하게 유지되는 가간섭성 광이 어떤 형태를 가진 물체에 부딪쳐 반사될 때는 그 반사된 광의 파면은 물체의 형태에 비례하여 달라진다. 즉 파면의 위상 변화가 물체의 형상에 따라 변화하게 된다. 그러므로 물체의 형상에 관한 좌표의 정보가 있으면, 형상에 따라 변화하는 파면의 위상변화는 기하광학적인 광선추적에 의해 쉽게 계산된다.

CGH는 식 (1)과 같이 정의되는데 앞서서 설명한 것과 같이 홀로그램의 위상(θ_H)으로부터 홀로그램의 강도(I_α)를 얻는 방법이다. 여기서 N 은 3차원 객체의 광원수를 뜻한다. θ_H 는 $kR_{\alpha j}$ 로 정의되고 이는 식 (2)와 같다. 여기서 k 는 참조파의 파수로 $2\pi/\lambda$ 로 정의되고 λ 는 $532nm$ 의 값을 사용한다. x_α 와 y_α 는 홀로그램내의 위치를 뜻하고 x_j , y_j , 및 z_j 는 3차원 객체의 위치를 나타낸다. p 는 픽셀의 크기를 나타내고 $10.4\mu m$ 의 값을 사용한다.

$$I_\alpha = \sum_j^N A_j \cos(\theta_H + \Phi_\alpha + \Phi_j) \quad (1)$$

$$R_{\alpha j} = \sqrt{(px_\alpha - px_j)^2 + (py_\alpha - py_j)^2 + z_j^2} \quad (2)$$

식 (2)는 Fresnel Approximation에 의해서 식 (3)으로 근사화된다[8].

$$R_{\alpha j} \cong z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2) \quad (3)$$

$kR_{\alpha j}$ 에 해당하는 항은 식 (4)와 같이 정리할 수 있다. 여기서 $x_{\alpha j}$ 와 $y_{\alpha j}$ 는 $x_\alpha - x_j$ 와 $y_\alpha - y_j$ 이다.

$$\begin{aligned} kR_{\alpha j} &= k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) \\ &= 2\pi(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) \\ &= 2\pi(\theta_Z + \theta_{XY}) \end{aligned} \quad (4)$$

식 (4)에 의해서 식 (1)은 식 (5)와 같이 나타낼 수 있다. 식 (5)와 같이 유도하는 이유는 코사인 함수는 2π 를 주기로 하여 반복되므로 $(\theta_Z + \theta_{XY})$ 의 결과에서 정수부분을 고려하지 않아도 되기 때문이다.

$$\begin{aligned} I_\alpha &= \sum_j^N A_j \cos(k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) + \Phi_\alpha + \Phi_j) \\ &= \sum_j^N A_j \cos(2\pi(\theta_Z + \theta_{XY}) + \Phi_\alpha + \Phi_j) \end{aligned} \quad (5)$$

소수부분의 연산만 수행함을 나타내는 연산자 $\text{mod}[]$ 를 도입한다면 식 (5)에서 θ_H 항은 식 (6)과 같이 유도할 수 있다. 식 (5)와 식 (6)에서 θ_H 항에 큰 차이는 없어 보이지만 정수부분의 연산이 없어지므로 곱셈과 덧셈 연산에서 상당히 많은 연산량이 감소한다.

$$\begin{aligned} kR_{\alpha j} &= k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) \\ &= \text{mod}\left[k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2))\right] \\ &= \text{mod}\left[2\pi(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2))\right] \\ &= \text{mod}\left[2\pi(\theta_Z + \theta_{XY})\right] \\ &= 2\pi \times \text{mod}\left[\theta_Z + \theta_{XY}\right] \end{aligned} \quad (6)$$

식 (6)을 이용하여 식 (5)을 재정리하면 식 (7)과 같다.

$$I_{\alpha} = \sum_j^N A_j \cos(2\pi \times \text{mod}[\theta_Z + \theta_{XY}] + \Phi_{\alpha} + \Phi_j) \quad (7)$$

식 (7)은 컴퓨터 프로그래밍에 의해서 수행되고 복원될 수 있다. 컴퓨터 프로그래밍에 의해서 뿐만 아니라 SLM을 이용하여 광학적으로 복원할 수도 있다. 이러한 과정을 그림 2에 나타냈다.

그림 2. CGH 생성 및 복원 과정
Fig. 2 CGH generation and reconstruction process

그림 3은 그림 2의 CGH 생성 및 복원 결과를 나타냈다. 그림 3(a)는 광원에 해당하는 원본 객체이고, 그림 3(b)는 생성된 홀로그램이다. 그림 3(b)에 참조파를 조사하면 그림 3(c)의 복원된 객체를 얻는다.

그림 3. CGH 및 복원 결과 (a) 원본 객체
(b) 디지털 홀로그램 (c) 복원된 객체
Fig. 3 CGH and reconstruction result (a) original object
(b) digital hologram (c) reconstructed object

III. 고정소수점 시뮬레이션

CGH는 다른 신호처리 기술들과 다르게 연산되는 변수들이 너무나 다른 범위를 갖는다. 즉 소수점이 거의 없고 정수부만 큰 값을 가지는 변수와 정수부는 거의 없으면서 소수부만을 큰 자리수를 가지는 변수 사이의 연산

이 빈번하다. 이러한 연산은 연산기의 크기만 증가시키면서 효율성은 떨어뜨린다. 따라서 하드웨어를 구현하기에 앞서서 중간 연산들 사이의 수 범위와 정밀도를 분석하여 구현하고자 하는 하드웨어의 비트너비(비스크기)를 정한다.

소프트웨어로 구현하는 경우에는 높은 정밀도를 갖고 연산을 수행하기 때문에 결과의 오차가 발생하지 않는다. 그러나 하드웨어로 구현할 경우에는 하드웨어 자원의 제한 때문에 정밀도에 한계를 갖는다. 고정소수점 연산 방법을 그림 5에 나타냈다. 정수와 소수로 이루어진 두 개의 연산자를 곱하는 경우를 고려하자. CGH 알고리즘에서 cosine 함수는 2π 단위로 반복한다. 즉 소수부만 사용되므로 그림 4와 같이 세 개의 부분곱이 발생한다.

그림 4. 코사인 함수의 위상을 위한 고정소수점 연산
Fig. 4 Fixed-point arithmetic for phase of cosine function

이들 부분곱을 모두 더하면 원하는 값이 된다. 고정소수점 시뮬레이션은 소수부분의 비트를 증가시켜가면서 CGH 알고리즘을 수행하여 결과를 확인한다. 비트너비가 0이면 정수만 사용하여 CGH를 수행하는 것이고, 1이면 소수가 표현할 수 있는 수는 0.5이다. 비트너비가 2이면 0.5와 0.25를 조합한 수까지 표현할 수 있다. 고정소수점 시뮬레이션 결과를 원래의 소프트웨어를 이용한 결과와 비교하여 완벽히 복원되는 비트수를 찾는다.

디지털 홀로그램과 객체에 대한 수치적인 해석과 시각적인 해석을 시도하고 결과를 분석한다. 수치적인 해석을 위해서는 peak signal-to-noise ratio (PSNR)과 normalized correlation(NC)을 이용한다.

(a)

(b)

그림 5. 홀로그램의 고정소수점 시뮬레이션
(a) PSNR (b) NC
Fig. 5 Fixed-point simulation for hologram
(a) PSNR (b) NC

디지털 홀로그램과 복원된 객체에 대한 PSNR 및 NC 결과를 그림 5와 6에 각각 나타냈다. 고정소수점을 위해 선택된 파라미터는 cosine 함수, Δ , π , 및 θ_z 이다. 결과에 영향을 미치는 중요한 요소를 선택한 것이다. 그림 5(a)의 PSNR 결과를 살펴보면 Δ 가 가장 많은 비트너비를 요구하는 것을 알 수 있다. 그림 5(b)의 NC 결과를 살펴보면 θ_z 가 가장 좋은 결과인 것을 알 수 있다. 그림 5(a)와 (b)를 종합적으로 고려하면 비트너비에 따른 PSNR과 NC 사이의 경향성은 일치하지 않다는 것을 알 수 있다. PSNR은 비교적 선형적으로 증가하는 반면에 NC는 특정한 비트 너비부터 급격히 증가하는 경향을 보인다. 일반적으로 30dB 이상이 되어야 시각적인 품질이 좋은 것으로 판단하기 때문에 PSNR 측면에서 cosine 함수는 약 2이상, Δ 는 약 25이상, π 는 1이상, 및 θ_z 는 3이상의 비트너비를 가져야 한다.

그림 6은 객체에 대한 고정소수점 연산 결과이다. 그림 6과 거의 유사한 경향성을 보이고 있다.

(a)

(b)

그림 6. 객체의 고정소수점 시뮬레이션
(a) PSNR (b) NC
Fig. 6 Fixed-point simulation for object
(a) PSNR (b) NC

IV. 시각적인 해석

그림 7부터 10에 객체의 복원 결과를 나타냈다. 그림들을 살펴보면 PSNR 및 NC 결과와 달리 비트너비가 작거나 비트너비가 큰 경우라 할지라도 복원된 객체의 결과는 시각적으로 거의 유사하다는 것을 확인할 수 있다. 코사인 함수의 경우에 1비트만 할당되어도 30비트가 할당된 경우와 거의 유사한 결과를 나타낼 수 있다. 또한 π 는 비트수가 결과에 거의 영향을 미치지 않는다는 것도 알 수 있다. 그러나 Δ 는 20비트 이상이 되어야만 원래의 객체와 유사해진다. 즉, 비트너비가 매우 민감하다는 것을 확인할 수 있다. θ_z 도 연산에서 제외시키더라도 결과에 크게 영향을 미치지 않는다는 것을 알 수 있다. 그러나 그림 7에서 10의 결과는 1m 거리에서 객체의 단면을 나타내는 것이기 때문에 전체적인 영향을 고려한다면 연산에서 제외시킬 수는 없을 것이다.

그림 7. 코사인 함수의 시각적 결과
(a) 0비트 (b) 1비트 (c) 15비트 (d) 30비트
Fig. 7 Visual result for cosine function
(a) 0bit (b) 1bits (c) 15bits (d) 30bits

그림 8. π 의 결과 (a) 0, (b) 10, (c) 20, (d) 29비트
Fig. 8 Visual result for π (a) 0, (b) 10, (c) 20, (d) 29 bits

결론적으로 고정소수점 연산을 이용하여 CGH를 위한 하드웨어의 비트수를 결정할 때에는 홀로그램 및 복원된 객체의 PSNR 및 NC와 함께 시각적인 관찰을 동시에 고려해야 한다. S/W와 동일한 수준의 H/W의 결과를 얻기 위해서는 표 1과 같은 비트 너비를 이용해야 한다. 표 1을 보면 Δ 의 경우에 소프트웨어와 동일한 결과를 보이기 위해서는 소수부분을 위해 51비트의 비트너비가 필요하다. 실제의 이러한 비트 너비로 하드웨어를 구현한다는 것은 비용 면에서 너무 비효율적이고, 어떤 경우에는 구

현이 불가능할 수도 있다. 그러나 수치적으로 완벽히 복원된 결과가 아니더라도 열화의 정도가 시각적으로 판단하기 어렵다면 그에 해당되는 비트 너비를 사용해도 무방하다. 따라서 복원된 객체를 수치적인 결과와 동시에 시각적으로 확인하여 최적화된 지점을 찾는다.

표 1. 정밀도 분석에 따른 비트수
Table. 1 Bit width by precision analysis

	Cosine	π	Δ	θ_z
Integer	2	3	0	22
Decimal	28	17	51	20

그림 9. Δ 의 시각적 결과 (a) 13, (b) 15, (c) 17, (d) 19, (e) 21, (f) 23, (g) 25, (h) 52비트
Fig. 9 Visual result for Δ (a) 13, (b) 15, (c) 17, (d) 19, (e) 21, (f) 23, (g) 25, (h) 52bits

그림 10. θ_Z 의 시각적 결과

(a) 0비트 (b) 2비트 (c) 3비트 (d) 34비트
Fig. 10 Visual result for θ_Z
(a) 0bit (b) 2bits (c) 3bits (d) 34bits

시각적인 특성을 고려해서 최적화된 비트 너비를 이용하는 것이 바람직하다. 표 2에는 각 항목별로 비트 제약을 준 다음에 고정소수점 시뮬레이션을 수행한 결과이다. 두 번째 행의 경우에는 코사인 함수를 8비트로 주고 다른 모든 항목은 double 형을 사용한 것이다. 표 3에는 각 항목들을 동시에 비트 제약을 가한 경우의 결과이다. 코사인 함수, π , 그리고 θ_Z 는 소수부분으로 8비트의 너비를 사용한다. 그리고 Δ 를 위해서는 32비트를 사용한다. 반드시 이 값들이 올바르고 유일한 선택은 아니다. 디지털 홀로그램의 사용분야와 하드웨어 비용에 따라서 선택할 수 있다. 이러한 비트 너비를 선택한 이유는 하드웨어의 연산기가 일반적으로 8비트 단위로 되어 있기 때문이다. 또한 이와 함께 객체의 PSNR이 비교적 우수하고 결과 영상이 원본과 비교할 때 크게 품질이 떨어지지 않기 때문이다.

표 2. 각 항목별 비트 제약에 따른 PSNR 결과
Table. 2 PSNR results according to bit constraint for each item

	비트수	홀로그램 PSNR	복원 객체 PSNR
Cosine	8	61.815	28.547
Delta	32	60.9589	28.396000
Pi	8	64.4586	29.841
ThetaZ	8	53.5612	27.073000

표 3. 비트 제약에 따른 PSNR 결과
Table. 3 PSNR results to bit constraint

	비트수	홀로그램 PSNR	복원 객체 PSNR
Cosine	8	52.7991	26.2477
Delta	32		
Pi	8		
ThetaZ	8		

V. 결론

본 논문에서는 ASIC, FPGA 및 GPU 등의 하드웨어 기반으로 컴퓨터 생성 홀로그램을 구현할 경우에 내부 시스템의 비트 너비를 분석하여 가이드라인을 제시하였다. 이 과정은 고정소수점 시뮬레이션을 통해서 수행되었고, 수치적인 결과와 시각적인 결과를 종합적으로 고려하였다. 이 결과들을 통하여 설계자는 응용 환경에 따라 시스템의 성능을 선택할 수 있다. 또한 하드웨어 구현을 위한 최적의 비트 너비도 보였고 이때 복원 객체의 PSNR은 약 26dB였다. 물론 홀로그램 및 홀로그램을 이용해 복원한 객체는 일반 영상들의 PSNR과는 조금 다른 경향을 보일 수 있다.

참고문헌

- [1] T. Motoki, H. Isono, and I. Yuyama, "Present Status of Three-Dimensional Television Research," Proc. IEEE 83(7): 1009-1021(July 1995).
- [2] J. K. Chung and M. H. Tsai, Three-Dimensional Holographic Imaging, John Wiley & Sons, Inc., 2002.
- [3] P. Hariharan, Basics of Holography, Cambridge University Press, May 2002.
- [4] Mark Lucente, "Interactive Computation of Holograms Using a Look-up Table", Journal of Electronic Imaging, vol. 2, #1, pp. 28-34, Jan. 1993.
- [5] H. Yoshikawa, S. Iwase, and T. Oneda, "Fast Computation of Fresnel Holograms employing Differences", Proceeding of SPIE, vol. 3956, 2000.
- [6] T. Shimobaba, T. Ito, "An efficient computational method suitable for hardware of computer-generated hologram with phase computation by addition",

- Computer Physics Communications, vol. 138, pp. 44-52, 2001.
- [7] N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie, "Computer generated holography using a graphics processing unit," Optics Express, Vol. 14, No. 2, January 2006.
- [8] L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holography using parallel commodity graphics hardware," Optics Express, Vol. 14, No. 17, August 2006.
- [9] Y. Pan, X. Xu, S. Solanki, X. Liang, R. Bin, A. Tanjung, C. Tan, and T.-C. Chong, "Fast CGH computation using S-LUT on GPU", Optics Express, vol. 17, No. 21, pp. 18543-18555, Oct. 2009.
- [10] Y.-Z. Liu, J.-W. Dong, Y.-Y. Pu, B.-C. Chen, H.-X. He, and H.-Z. Wang, "High-speed full analytical holographic computations for true-life scenes", Optics Express, vol. 18, no. 4, pp. 3345-3351, Feb. 2010.
- [11] T. Shimobaba, T. Ito, N. Masuda, Y. Ichihashi, and N. Takada, "Fast calculation of computer-generated-hologram on AMD HD5000 series GPU and OpenCL", Optics Express, vol. 18, no. 10, pp. 9955-9960, May. 2010.
- [12] T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba, and T. Sugie, "Special-Purpose computer HORN-5 for a real-time electroholography," Optics Express, Vol. 13, No. 6, March 2005.
- [13] Y. Ichihashi, H. Nakayama, T. Ito, N. Masuda, T. Shimobaba, A. Shiraki, and T. Sugie, "HORN-6 special-purpose clustered computing system for electroholography", Optics Express, vol. 17, no. 16, pp. 13895-13903, Aug, 2009
- [14] Y.-H. Seo, H.-J. Choi, J.-S. Yoo, and D.-W. Kim, "An architecture of a high-speed digital hologram generator based on FPGA", Journal of Systems Architecture, Vol. 56. pp. 27-37, Dec. 2009.
- [15] Young-Ho Seo, Hyun-Jun Choi, Ji-Sang Yoo, and Dong-Wook Kim, "Cell-based Hardware Architecture for full-parallel generation algorithm of digital holograms", Optics Express, Vol. 19, pp. 8750~8773, Apl. 20

저자소개

서영호(Young-Ho Seo)

1999년 2월 광운대학교 전자재료
공학과 졸업(공학사)
2001년 2월 광운대학교 일반대학원
졸업(공학석사)

2004년 8월 광운대학교 일반대학원 졸업(공학박사)
2008년 3월 ~ 현재 광운대학교 교양학부 부교수
※관심분야: 실감미디어, 2D/3D 영상 신호처리,
디지털 홀로그래프

김동욱(Dong-Wook Kim)

1983년 2월 한양대학교 전자공학과
졸업(공학사)
1985년 2월 한양대학교 공학석사
1991년 9월 Georgia 공과대학
전기공학과(공학박사)

1992년 3월 ~ 현재 광운대학교 전자재료공학과 정교수
2009년 3월 ~ 현재 광운대학교 실감미디어 연구소 소장
※관심분야: 3D 영상처리, 디지털 홀로그래프, 디지털
VLSI Testability, VLSI CAD