

일반논문-11-16-1-06

디지털 홀로그램의 고속 생성을 위한 병렬화 알고리즘 및 셀 기반의 하드웨어 구조

서영호^{a)}, 최현준^{a)}, 유지상^{a)}, 김동욱^{a)†}

A New Parallelizing Algorithm and Cell-based Hardware Architecture for High-speed Generation of Digital Hologram

Young-Ho Seo^{a)}, Hyun-Jun Choi^{a)}, Ji-Sang Yoo^{a)}, and Dong-Wook Kim^{a)†}

요 약

본 논문에서는 고속으로 홀로그램을 생성하기 위해 새로운 컴퓨터 생성 홀로그램(computer-generated hologram, CGH) 수식을 제안하고, 셀 기반의 VLSI(very large scale integrated circuit) 구조를 제안하였다. 기본 CGH 수식에서 가로 또는 세로 방향의 연산 규칙을 찾아낸 후 가로 또는 세로 방향의 홀로그램 화소를 병렬적으로 구할 수 있는 수식을 유도하였다. 제안한 수식을 바탕으로 초기 파라미터 연산기(initial parameter calculator)와 업데이트-위상 연산기(update-phase calculator)로 구성된 CGH 셀의 구조를 제안하고 하드웨어로 구현하였다. 수식의 변형을 통해서 하드웨어를 간략화 시킬 수 있었고, CGH의 확장을 통해 가로 방향으로 병렬화시킬 수 있는 하드웨어 구조도 보였다. 실험에서는 하드웨어에 사용된 자원을 분석하였다. CGH 커널과 프로세서의 구조는 이전 연구에서 사용된 플랫폼을 그대로 사용하였다.

Abstract

This paper proposes a new equation to calculate computer-generated hologram (CGH) in a high speed and its cell-based VLSI (very large scale integrated circuit) architecture. After finding the calculational regularity in the horizontal or vertical direction from the basic CGH equation, we induce the new equation to calculate the horizontal or vertical hologram pixel values in parallel. We also propose the architecture of the CGH cell consisting of a initial parameter calculator and update-phase calculator(s) on the basis of the equation and implement them in hardware. Also we show a hardware architecture to parallelize the calculation in the horizontal direction by extending CGH. In the experiments we analyze the used hardware resources. These analyses makes it possible to select the amount of hardware to the precision of the results. Here, for the CGH kernel and the structure of the processor, we used the platform from our previous works.

Keyword : digital hologram, computer-generated hologram, VLSI, hardware design, parallel architecture

a) 광운대학교
Kwangwoon University

b) 안양대학교
Anyang University

† 교신저자 : 김동욱 (dwkim@kw.adc.kr)

※ 본 연구는 지식경제부, 방송통신위원회 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음. [KI002058, 대화형 디지털 홀로그램 통합서비스 시스템의 구현을 위한 신호처리 요소 기술 및 SoC 개발]
· 접수일(2010년10월11일), 수정일(2010년12월30일), 게재확정일(2010년12월30일)

1. 서론

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]. 여기서는 HPO(horizontal-parallax-only) 방식의 CGH 기법을 이용해 디지털 홀로그램을 생성하는 연구를 수행하였다. 이 연구에서는 LUT(Look-up Table)방식과 parallel super-computer를 이용해 10,000 point의 light source 영상으로부터 1초에 한 장씩 디지털 홀로그램(해상도: 6M)을 생성하였다. 일본 Nihon대학의 Yoshikawa교수는 root연산을 Taylor 전개를 통해 변형한 후 CGH 수식의 근사화를 통해 고속화 알고리즘을 정리하였다^[5]. 이 연구는 최근 연구되고 있는 고속 CGH 알고리즘들의 이론적 토대가 된다는 점에서는 상당히 가치가 있지만, 연산속도 측면에서는 큰 의미가 없다. 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 진행되

어 왔다^[7-14]. 이러한 연구들은 GPU 기반의 소프트웨어 방식^[7-11]과 FPGA 기반의 하드웨어 칩 구현 방식^{[6][12-14]}으로 나누어진다. GPU를 이용한 방식은 FPGA 기반의 방식에 비해서 구현이 비교적 쉽고 개발기간이 짧은 장점이 있다. FPGA를 이용하면 구현과정이 매우 복잡하고 개발기간이 오래 걸린다는 단점이 있다. 또한 한번 구현하면 구조를 변경하거나 성능을 개선하기 어렵다. 그러나 GPU 방식에 비해서 성능은 수십에서 수백배 가량 높은 성능을 갖는다. 특히 GPU에 대한 연구가 최근에 활발히 이루어지고 있다. 싱가포르대^[9]는 CGH 수식을 exponent 함수를 이용한 복소형태로 변환한 후에 연산을 분리하는 알고리즘을 제안하였다. 분리된 항을 각각 LUT로 만든 후에 연산을 고속화시켰고, 이를 nVidia의 GPU로 구현하였다. 1,000(1K)개의 object point를 갖는 객체에 대해 1024×768크기의 홀로그램을 0.3초당 한 장씩 생성할 수 있었다. 중국 Zhongshan 대학의 Wang^[10]은 3D mesh-model을 기반으로 GPU를 이용해 CGH를 수행한 연구를 발표했다. 또한 일본 Chiba대의 Shimobaba^[11]는 AMD의 GPU를 기반으로 하여 이전 연구에서 제안한 알고리즘^[6]을 사용하면서 GPU 프로그래밍 기법을 활용하여 고속화하였고, HD크기의 홀로그램을 0.31초당 한 장씩 생성할 수 있었다. [12]에서는 4개의 Xilinx FPGA(XC2VP70)를 사용하는 전용 PCB 보드를 제작하여 Fresnel Transform CGH를 구현하였다. 홀로그램의 x축 해상도만큼의 단위 연산기를 병렬로 배열(1,408개)하는 구조를 가지고 166MHz의 클럭 주파수에서 한 프레임의 홀로그램을 0.0679초에 생성할 수 있다. 최근에는 CGH를 연산하기 위한 전용 연산 시스템인 HORN-6 특수 컴퓨터가 제안되기도 하였다^[13]. 또한 100% 파이프라인(pipeline) 구조를 기반으로 하는 CGH 프로세서가 제안되었다^[14]. Fresnel 변환을 수행하기 위한 CGH Cell의 하드웨어 구조를 제안한 후에 이를 확장하여 CGH Kernel을 구성하였고, 이를 다시 확장하여 CGH 프로세서를 구현하였다. [14]의 하드웨어는 [12]보다 최대 87.32%의 높은 성능을 갖는다.

[12]와 [14]의 논문에서 사용하고 있는 홀로그램의 최대 크기는 1408×1050이다. 픽셀 크기가 10 μ m의 SLM을 사용하면 홀로그램의 크기는 1.408×1.050cm이다. 광학적인 확대를 고려한다고 해도 너무나 작은 홀로그램이기 때문에

CGH 연산을 위해 지금보다 훨씬 높은 성능을 갖는 하드웨어가 필요하다.

본 연구팀은 이전 연구^[14]에서 제안된 하드웨어 구현을 위한 CGH 연산 기법을 재구성한 후 파이프라인 기법을 적용하여 고성능의 CGH 프로세서를 제안하였다. 본 연구에서는 이전 연구에서 제안된 것보다 더욱 우수한 성능을 가질 수 있게 하기 위하여 확장이 용이한 변형된 CGH 연산 기법을 제안한다. 또한 이를 바탕으로 고성능의 하드웨어 구조를 새롭게 제안하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 CGH의 원리와 하드웨어 구현을 위해 수정된 CGH 알고리즘을 설명한다. 3장에서는 제안한 하드웨어 구조를 설명하고 4장에서는 구현결과를 보이고 기존 연구와 비교한다. 마지막으로 5장에서는 결론을 맺는다.

II. 컴퓨터 생성 홀로그램

1. CGH 알고리즘

홀로그램은 광학계를 이용하여도 취득할 수 있지만 광학계 자체를 수학적으로 모델링한 연산에 의해서 구할 수도 있다. 이러한 수학적 연산을 통해 얻어진 홀로그램을 컴퓨터 생성 홀로그램(computer-generated hologram, CGH)이라고 한다. 여러 종류의 CGH가 있지만 본 논문에서는 “위상(phase)” 방식을 사용하는데, 이것은 객체에서 CCD로 입사되는 파에서 위상 성분만을 이용하여 홀로그램을 생성하는 것으로, 그 증명^[3]은 본 논문에서 다루지 않는다.

파면의 간격이 시간과 공간에 걸쳐 일정하게 유지되는 가간섭성(coherent) 광이 어떤 형태를 가진 물체에 부딪쳐 반사될 때 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 앞서 설명한 것과 같이 홀로그램의 위상으로부터 홀로그램의 강도(I_α)를 얻는 방

법이다. 여기서 N은 3차원 객체의 광원수를 뜻한다. k 는 참조파의 파수로 $2\pi/\lambda$ 로 정의되고 λ 는 사용된 파의 파장을 나타낸다. 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 나타낸다.

$$I_\alpha = \sum_j^N A_j \cos(k \sqrt{(px_\alpha - px_j)^2 + (py_\alpha - py_j)^2 + z_j^2} + \Phi_\alpha + \Phi_j) \quad (1)$$

식 (1)에서 제곱근은 $x_{\alpha j}$, $y_{\alpha j} \ll z_j$ 의 조건인 경우에 식 (2)와 같이 Fresnel 근사를 통해서 근사될 수 있다^[5]. $x_{\alpha j}$ 와 $y_{\alpha j}$ 는 각각 $x_{\alpha j} = x_\alpha - x_j$ 및 $y_{\alpha j} = y_\alpha - y_j$ 로 정의된다.

$$\sqrt{(px_\alpha - px_j)^2 + (py_\alpha - py_j)^2 + z_j^2} \cong z_j + \frac{p^2}{2z_j} (x_{\alpha j}^2 + y_{\alpha j}^2) \quad (2)$$

식 (2)를 이용하여 식 (1)을 다시 정리하면 식 (3)과 같다.

$$I_\alpha = \sum_j^N A_j \cos(k(z_j + \frac{p^2}{2z_j} (x_{\alpha j}^2 + y_{\alpha j}^2)) + \Phi_\alpha + \Phi_j) \quad (3)$$

2. 수정된 CGH 알고리즘

본 절에서는 앞 절에서 설명한 CGH 알고리즘의 효율을 높이기 위해 제안된 방법을 소개한다^[6]. 먼저 식 (3)을 θ_Z 와 θ_{XY} 를 이용하여 정의하면 식 (4)와 같고, 각각은 식 (5)와 식 (6)으로 정의된다.

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

$$\theta_Z(z_j) = \frac{z_j}{\lambda} \quad (5)$$

$$\theta_{XY}(x_{\alpha j}, y_{\alpha j}, z_{\alpha j}) = \frac{p^2}{2\lambda z_j} (x_{\alpha j}^2 + y_{\alpha j}^2) \quad (6)$$

홀로그래프 평면의 한 행에서 제일 첫 번째 화소위치로부터 d 번째 화소위치의 θ_{XY} 를 구하면 식 (7)과 같다. 식 (7)에서 $\frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)$ 는 초기에 연산되는 항에 해당하고, 첫 번째 위치에서의 θ_{XY} 와 차이는 $\frac{p^2}{2\lambda z_j}(2dx_{\alpha j} + d^2)$ 이다. 즉 가로방향으로 이동함에 따라서 그 만큼의 값이 차이 나는 것이다.

$$\begin{aligned} & \theta_{XY}(x_{\alpha j} + d, y_{\alpha j}, z_{\alpha j}) \\ &= \frac{p^2}{2\lambda z_j}((x_{\alpha j} + d)^2 + y_{\alpha j}^2) \\ &= \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + 2dx_{\alpha j} + d^2 + y_{\alpha j}^2) \\ &= \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2) + \frac{p^2}{2\lambda z_j}(2dx_{\alpha j} + d^2) \end{aligned} \quad (7)$$

이 수정된 CGH 알고리즘은 하나의 가로줄 단위로 연산이 이루어진다. 먼저, 가로줄의 첫 번째 홀로그래프 위치($d=0$)에 대해 연산을 수행하고 그 이후의 위치($d>0$)에 대해서는 가로줄이 끝날 때 까지 이전 위치에서 연산된 결과(Γ_{d-1})에 일부 값($\Gamma_1 + (d-1)\Delta$)을 보정하여 해당 위치에서의 홀로그래프 값(Γ_d)을 구한다. 가로줄의 첫 번째 위치에서 연산되어야 하는 과정은 식 (8)과 같다.

$$I_\alpha = \sum_j^N A_j \cos \left(2\pi \left(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j} (x_{\alpha j}^2 + y_{\alpha j}^2) \right) + \Phi_\alpha + \Phi_j \right) \quad (d=0) \quad (8)$$

($d=1$)의 위치부터 사용되는 연산식은 식 (9)과 같다. 식 (9)에서 Γ_d 는 식 (10)과 같고, Γ_1 와 Δ 는 식 (11) 및 식 (12)와 같다^{[8][9]}.

$$I_\alpha = \sum_j^N A_j \cos \left(2\pi \left(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j} (x_{\alpha j}^2 + y_{\alpha j}^2) + \Gamma_d \right) + \Phi_\alpha + \Phi_j \right) \quad (d \geq 1) \quad (9)$$

$$\Gamma_d = \Gamma_{d-1} + \Gamma_1 + (d-1)\Delta, \quad (d \geq 1) \quad (10)$$

$$\Gamma_1(x_{\alpha j}, z_j) = \frac{p^2}{2\lambda z_j}(2x_{\alpha j} + 1) \quad (11)$$

$$\Delta = \frac{p^2}{\lambda z_j} \quad (12)$$

식 (8)과 (9)를 이용하여 객체의 첫 번째 광원으로 중간 홀로그래프를 구하여 저장하고, 두 번째 광원으로 두 번째 중간 홀로그래프를 구한 후 첫 번째 중간 홀로그래프와 더한다. 객체를 구성하는 광원의 수 만큼 이러한 과정을 반복하여 최종적인 홀로그래프를 생성한다.

III. 제안한 구조

1. 병렬화 수식의 유도

본 절에서는 제안하고자 하는 연산 방식에 대해서 설명한다. 앞서 설명한 CGH 생성 수식을 새롭게 재구성하여 병렬 연산에 적합하게 한다. Γ_d 는 식 (13)으로 정의할 수 있고, d 값을 증가시키면서 Γ_d 를 구하면 식 (14)에서 (17)과 같이 Γ_1 과 $\frac{p^2}{\lambda z_j}$ 에 대한 연산으로 정리할 수 있다. 이를 이용하여 임의의 위치에서의 Γ_d 를 구하면 식 (18)과 같은 일반항을 유도할 수 있다.

$$\Gamma_d(x_{\alpha j}, z_j) = \frac{p^2}{2\lambda z_j}(2dx_{\alpha j} + d^2) \quad (13)$$

$$\Gamma_1(x_{\alpha j}, z_j) = \frac{p^2}{2\lambda z_j}(2x_{\alpha j} + 1) = 1\Gamma_1 + 0\Delta \quad (14)$$

$$\begin{aligned} \Gamma_2(x_{\alpha j}, z_j) &= \frac{p^2}{2\lambda z_j}(4x_{\alpha j} + 4) \\ &= 2\frac{p^2}{2\lambda z_j}(2x_{\alpha j} + 1) + 2\frac{p^2}{2\lambda z_j} \\ &= 2\Gamma_1 + 1\Delta \end{aligned} \quad (15)$$

$$\begin{aligned} \Gamma_3(x_{\alpha j}, z_j) &= \frac{p^2}{2\lambda z_j}(6x_{\alpha j} + 9) \\ &= 3\frac{p^2}{2\lambda z_j}(2x_{\alpha j} + 1) + 6\frac{p^2}{2\lambda z_j} \\ &= 3\Gamma_1 + 3\Delta \end{aligned} \quad (16)$$

$$\begin{aligned} \Gamma_4(x_{\alpha j}, z_j) &= \frac{p^2}{2\lambda z_j} (8x_{\alpha j} + 16) \\ &= 4 \frac{p^2}{2\lambda z_j} (2x_{\alpha j} + 1) + 12 \frac{p^2}{2\lambda z_j} \\ &= 4\Gamma_1 + 6\Delta \end{aligned} \quad (17)$$

$$\Gamma_d(x_{\alpha j}, z_j) = \frac{p^2}{2\lambda z_j} (2dx_{\alpha j} + d^2) = d\Gamma_1 + \frac{d(d-1)}{2} \Delta \quad (18)$$

하드웨어의 효율성을 고려한다면 식 (18)은 식 (19)와 같이 변형하는 것이 유리하다. 두 식이 동일한 것으로 보이지만 하드웨어 구현 측면에서는 차이점을 갖는다. 하드웨어의 측면에서 식 (18)은 3개의 곱셈기와 1개의 덧셈기가 필요하지만 식 (19)는 2개의 곱셈기와 1개의 덧셈기가 필요하다. 즉, 곱셈기의 개수가 1개 감소한다. 1개의 곱셈기 자체의 감소는 중요하지 않게 생각될 수 있다. 그러나 CGH 셀이 홀로그래프의 행의 개수(예를 들어 1280개)만큼 사용된다. 이를 고려한다면 중요한 요소로 고려될 수 있다. $(d-1)$ 연산은 단순한 카운팅 동작이고 $\frac{1}{2}$ 연산은 하드웨어 배선에 의한 쉬프트 동작이다.

$$\Gamma_d = d \left\{ \Gamma_1 + \frac{1}{2} (d-1) \Delta \right\} \quad (19)$$

식 (19)는 식 (10)과 동일한 결과를 가지는 식이지만 식 (10)과 달리 초기값이 주어진 이후에는 임의의 위치에서 홀로그래프 계수 생성이 가능하다는 장점을 갖는다. 이러한 장

점은 하드웨어 구현 시 다양한 병렬화를 가능하게 해줄 수 있다. 뿐만 아니라 홀로그래프 평면에서 임의의 영역에 대한 홀로그래프를 선별적으로 구할 수도 있어서 다양한 활용이 가능하다. 예를 들어 수평방향으로 다수 개의 홀로그래프 구성 성분을 동시에 연산할 수 있다. 그림 1(a)와 같이 홀로그래프 평면의 한 열을 연산하고자 할 경우에 그림 1(b)와 같이 요구되는 성능에 따라서 병렬화를 시킬 수가 있다.

2. 제안한 하드웨어 구조

본 장에서는 새롭게 유도된 CGH 수식을 바탕으로 하여 새로운 하드웨어 구조를 제안한다. 하드웨어로 구현하기 위해서 식 (9)를 식 (20)으로 재정의한다. 식 (20)에서 처음에 계산된 후 동일한 값을 유지하는 부분과 변화되는 부분을 구분한다. 각각은 초기항(i_{init})과 업데이트항(i_{update})이라 부르고 식 (21)과 (22)로 정의한다^[14].

$$\begin{aligned} I_\alpha &= \sum_j^N A_j \cos(2\pi(\theta_Z + \theta_{XY,d=0} + \Gamma_d)) \\ &= \sum_j^N A_j \cos(2\pi(\theta_Z + \theta_{XY,d=0} + d\Gamma_1 + \frac{1}{2}d(d-1)\Delta)) \end{aligned} \quad (20)$$

$$i_{init}(x_{\alpha j}, y_{\alpha j}, z_j) = \{\theta_Z + \theta_{XY,d=0}, \Gamma_1, \Delta\} \quad (21)$$

$$i_{update}(i_{init}(), d) = \{\theta_H = \theta_Z + \theta_{XY,d=0} + \Gamma_d\} \quad (22)$$

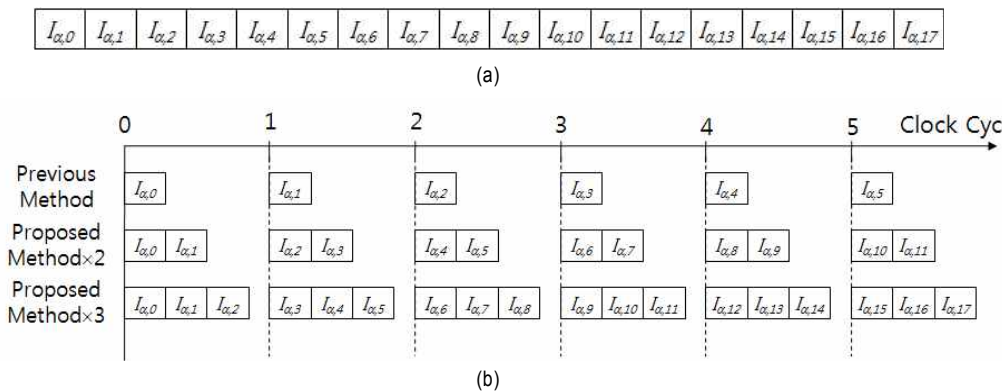


그림 1. 하드웨어 구현을 위한 연산 방식
Fig. 1. Operational method for hardware development

식 (21)으로 부터 객체 내의 하나의 광원에 의한 중간 홀로그래밍 내의 한 점은 식 (22)로 정의된다. 식 (21)의 초기항은 홀로그래밍의 한 열에 대해서 처음 한 화소에 대해서만 연산되고 그 이후부터는 사용되지 않는다. 따라서 식 (21)은 식 (22)와 다른 동작 특성을 갖는다. 하드웨어 구현을 고려한다면 식 (21)에서 θ_z 와 Δ 를 LUT로 만드는 것이 바람직하다. 두 개의 항 모두 z_j 를 변수로 갖는 연산이다. z_j 는 거리를 256단계로 나누는 깊이 정보이다. 따라서 θ_z 와 Δ 는 256개의 주소를 가지는 LUT1으로 통합하고 하드웨어에 내장시킨다. 만일에 z_j 가 너무 많은 종류의 값을 갖는

다면 룩업테이블로 만드는 것이 더욱 하드웨어 자원과 효율에 있어서 단점이 될 수도 있다. 또한 식 (22)에서도 하드웨어 구현을 고려한다면 코사인 함수 연산은 간단한 룩업테이블로 처리할 수 있다. 따라서 식 (20)에서 누적되는 항을 식 (23)과 같이 표현할 수 있다. 식 (23)은 LUT2의 결과를 A_j 로 곱하는 것으로 매우 단순한 과정이다.

$$I_{x_j}(A_j, i_{update}()) = A_j \cos(2\pi i_{update}()) = A_j \cos(2\pi \theta_H) = A_j LUT2(\theta_H) \quad (23)$$

그림 2(a)에 식 (21)의 초기항을 연산하기 위한 초기항

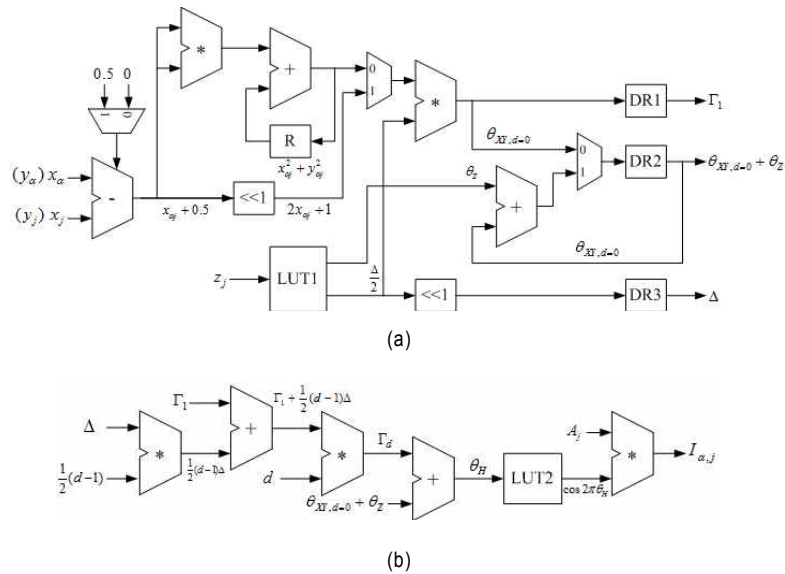


그림 2. CGH 셀의 세부 구조 (a) 초기 파라미터 연산기 (b) 업데이트-위상 연산기
Fig. 2. Detailed architecture of CGH cell (a) initial parameter calculator (b) update-phase calculator

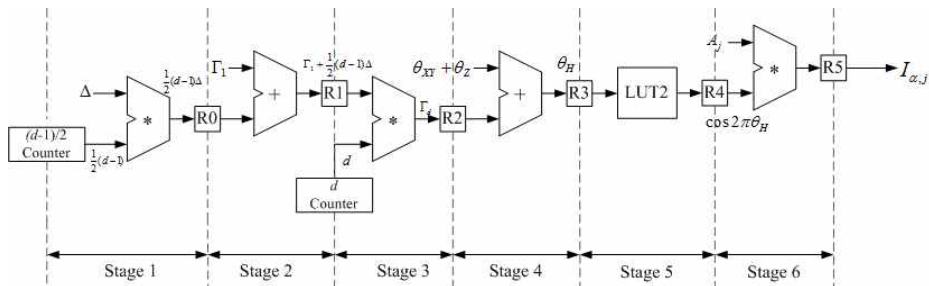


그림 3. 파이프라이닝 업데이트-위상 연산기의 구조
Fig. 3. The pipelined architecture of the update-phase calculator

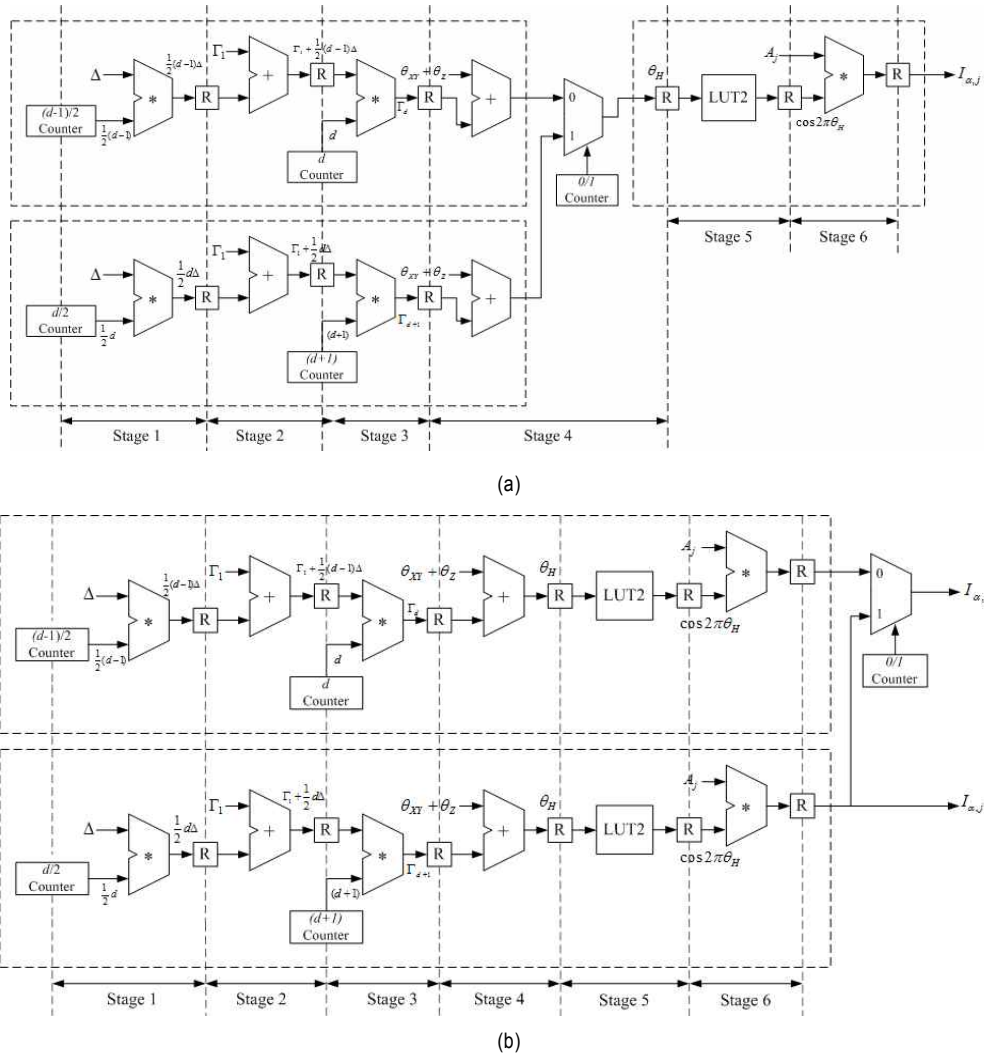


그림 4. 화소 단위의 병렬화를 위한 업데이트-위상 연산기의 확장, (a) 한쪽 방향으로만 확장 가능한 구조, (b) 양쪽방향으로 확장 가능한 구조
 Fig. 4. Expansion of update-phase calculator for pixel-based parallelization, (a) structure to extend only one direction, (b) structure to extend both direction.

연산기의 구조를 나타냈고, 그림 2(b)에는 식 (22)와 식 (23)을 연산하기 위한 업데이트-위상 연산기의 구조를 나타냈다. 초기항에서 구해지는 3가지 값들(Γ_1 , $\theta_Z + \theta_{XY, d=0}$, Δ)은 각각의 데이터 레지스터(DR1, DR2, DR3)에 저장되고, 업데이트항을 연산하는데 이용된다.

CGH 셀의 내부 단위 연산시간을 바탕으로 그림 3과 같이 CGH 셀을 파이프라인화하였다. 총 6 단계의 파이프라

인 단계를 가지므로 6 클록의 대기지연시간 이후에 한 클록 당 하나의 결과를 출력할 수 있다. 그림 3에서 R이 파이프라인 레지스터에 해당한다.

앞서 설명한 것과 같이 식 (17)과 식 (18)의 특성은 첫 번째 위치의 홀로그램 화소가 연산된 이후에 나머지 홀로그램 화소들은 병렬적으로 연산을 할 수 있다는 것이다. 만일 두 개의 화소 위치에 대해서 병렬적으로 동시에 연산을

하고자 한다면 그림 4와 같이 확장할 수 있다. 그림 4(a)는 LUT2를 공유하여 하드웨어의 양을 다소 감소시킬 수 있다. 반면에 그림 4(b)는 하드웨어의 양은 다소 증가하지만 출력을 직렬 및 병렬로 조절할 수 있는 장점이 있다.

3. CGH 프로세서

본 논문에서 제안된 CGH 셀의 구조를 확장하여 CGH 커널로 구현된다. 또한 CGH 커널과 함께 Input Interface, Output Interface, Main Controller, SDRAM Controller, DMA 등으로 CGH 프로세서가 구성된다. 본 논문에서 사용된 CGH 커널 구조와 CGH 프로세서의 구조는 [14]에서 사용한 구조를 그대로 사용한다^[14].

IV. 구현된 하드웨어

제안한 하드웨어는 VHDL을 이용하고 Altera사의 FPGA 환경을 이용하였다. VHDL의 설계는 Quartus II 10.0을 이용하였고, 시뮬레이션은 Modelsim 6.5e를 사용하였다. 표 1에 이전 연구^[14]와 새롭게 제안한 CGH 셀의 하드웨어 자원을 비교하였다. 제안한 CGH 셀은 기존의 하드웨어보다 더 작은 자원을 사용하면서도 가로방향의 병렬 연산이 가능한 장점을 갖는다.

한 프레임의 홀로그램을 계산하는데 걸리는 시간은 식 (24)로 정의할 수 있다. 제안한 하드웨어는 홀로그램 수직 해상도만큼의 CGH 셀수를 갖고, 100% 파이프라인 구조로 되어 있다. 따라서 초기 대기지연시간을 거친 후에 홀로그램 수평 해상도만큼의 클럭이 소요되면 한 장의 홀로그램이 생성될 수 있다.

$$CGH\ Rate = (Hologram\ Width \times Light\ Source + Initial\ Latency) \times Clock\ Period \quad (24)$$

식 (24)에 따라서 10K개의 광원을 갖는 객체에 대해서 1920×1080(HD)크기의 홀로그램 1장을 생성하는 데에는 63ms가 소요된다. 가로 방향으로 2개의 홀로그램 화소를 병렬로 연산하는 구조를 적용한다면 31.5ms가 소요되고, 3개에 대해서는 21ms가 소요된다.

표 1에서 [12]보다 [14]가 더 좋은 결과를 나타내므로 [14]와 제안한 방식의 성능을 비교할 수 있다. 먼저 [14]에 비해서 제안한 CGH 셀은 파이프라인화가 잘 되어 있기 때문에 더 높은 주파수에서 동작할 수 있다. 또한 [14]에 비해서 대기지연 시간이 짧아 성능이 더욱 우수할 수 있다. 제안한 하드웨어가 더 큰 홀로그램을 더 빠른 시간에 생성할 수 있다. 그러나 이것이 제안한 회로의 가장 큰 특징은 아니다. 가장 큰 장점은 [12]와 [14]의 방식에 비해서 병렬화가 훨씬 용이하다는 것이다.

표 1. 구현 결과의 비교
Table 1. Comparison of implementing results

Item	GPU		FPGA					
	[9]	[11]	[12]	[14]	Proposed			
Object Point	1,000	1,000	10,000			10,000		
Hologram Size	1,024×768	1,920×1,080	1,408×1,050			1,920×1,080	1,408×1,050	
Frequency (MHz)	-	-	166MHz			166MHz		
$\frac{Time(s)}{CGH}$	0.03	0.031	0.0679	0.0232	0.0159	0.065	0.016	0.021
$\frac{Hologram}{Time(s)}$	33.3	32.2	14.73	43.10	62.90	15.37	61.48	47.16
Parallel Unit Cells	nVidia	AMD	1,408	5,632	5,632	1,920	7,680	5,632

V. 결 론

본 논문에서는 새로운 CGH 수식을 유도하였고, 이 수식을 바탕으로 하여 고성능의 CGH 생성 하드웨어 구조를 제안하였다. 제안한 CGH 수식은 가로 방향으로 병렬 연산이 가능한 특성을 갖는다. 열 방향으로 연산을 하기 위해 필요한 초기 데이터들을 구한 다음에는 이 값을 이용하여 동일한 열에 있는 임의의 위치에서의 홀로그램 화소값을 구할 수 있다. 이 수식은 전용 칩셋이나 GPU로 구현할 경우에는 다양한 형태로 병렬 연산이 가능하게 하므로 CGH의 성능을 향상시킬 수 있게 해 준다. 제안한 수식을 바탕으로 CGH 셀을 제안하였다. CGH 셀은 초기 파라미터 연산기와 업데이트-위상 연산기로 구성된다. 업데이트-위상 연산기에 병렬화 기법을 적용할 경우에 다양한 병렬화가 가능하고 그 만큼 속도가 향상될 수 있다. 여기에 기존 연구들에서 사용되고 있는 수직방향의 병렬화 기법과 홀로그램 영역을 분할하는 영역 기반의 병렬화를 함께 쓴다면 더욱 고성능을 갖는 CGH 연산기를 구현할 수 있다. 본 논문은 CGH 셀의 구조에 초점을 맞추었고, CGH 셀을 이용한 CGH 커널과 CGH 프로세서는 이전 연구에서 제안된 하드웨어를 그대로 사용하였다. 기존 연구와 비교할 때 더욱 작은 하드웨어 자원을 사용하면서도 더 높은 성능을 보였고, 가로 방향의 병렬화가 가능하여 성능을 선형적으로 증가시킬 수 있는 장점도 갖는다. 본 연구는 실사 및 컴퓨터 그래픽을 이용한 CGH 응용 분야에 다양한 가이드라인과 정보를 제공할 수 있을 것으로 예상된다.

참 고 문 헌

- [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] W. G. Joseph, Introduction to Fourier Optics (3rd edition), Roberts & Company, 2005.

저 자 소 개



서 영 호

- 1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)
- 2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
- 2003년 6월 ~ 2004년 6월 : 한국전기연구원 연구원
- 2005년 9월 ~ 2008년 2월 : 한성대학교 조교수
- 2008년 3월 ~ 현재 : 광운대학교 교양학부 조교수
- 주관심분야 : 2D/3D 영상 및 비디오 처리, 디지털 홀로그래프, SoC 설계, 워터마킹/암호화



최 현 준

- 2003년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2005년 2월 : 광운대학교 일반대학원 졸업(공학석사)
- 2009년 2월 : 광운대학교 일반대학원 졸업(공학박사)
- 2009년 3월 ~ 2010년 2월 : 광운대학교 연구교수
- 2010년 3월 ~ 현재 : 안양대학교 정보통신공학과 조교수
- 주관심분야 : 영상압축, 워터마킹, 암호화, FPGA/ASIC 설계, Design Methodology



유 지 상

- 1985년 2월 : 서울대학교 전자공학과 졸업(공학사)
- 1987년 2월 : 서울대학교 대학원 전자공학과 졸업(공학석사)
- 1993년 5월 : Purdue 대학교 전기공학과 졸업(Ph.D.)
- 1993년 9월 ~ 1994년 8월 : 현대전자산업(주) 산전연구소 선임연구원
- 1994년 9월 ~ 1997년 8월 : 한림대학교 전자공학과 조교수
- 1997년 9월 ~ 현재 : 광운대학교 전자공학과 교수
- 주관심분야 : 3D 영상처리, 웨이블릿 기반 영상처리, 영상압축, 영상인식, 비선형 신호처리



김 동 욱

- 1983년 2월 : 한양대학교 전자공학과(공학사)
- 1985년 2월 : 한양대학교 공학석사
- 1991년 9월 : Georgia공과대학 전기공학과 (공학박사)
- 1992년 3월 ~ 현재 : 광운대학교 전자재료공학과 정교수 신기술 연구소 연구원
- 2000년 3월 ~ 2001년 : 12월인티스닷컴(주) 연구원.
- 2009년 3월 ~ 현재 : 광운대학교 실감미디어연구소 연구소장
- 2006년 3월 ~ 현재 : (사)실감미디어산업협회 이사
- 주관심분야 : 3D 영상처리, 디지털 홀로그래프, 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication