JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 4 : 820~828 Apr. 2017

드뷔시 필터를 이용한 고속 프레넬릿 변환의 구현

서영호[1] · 이윤혁[2] · 김동욱[2*]

Implementation of High-Speed Fresnelet Transform using Daubechies's Filter

Young-Ho Seo[1] · Yoon-Hyuk Lee[2] · Dong-Wook Kim[2*]

[1]Ingenium College of Liberal Arts, Kwangwoon University, Seoul 01897, Korea
[2*]Department of Electronic Materials Engineering, Kwangwoon University, Seoul 01897, Korea

요 약

차세대 영상 시스템인 디지털 홀로그램을 서비스하기 위해서는 다양한 측면에서 신호처리 기술이 필요하다. 우리가 주로 사용하는 2차원 및 3차원 자연 영상을 처리하는데 있어서 가장 많이 사용되는 영상처리 도구는 변환이다. 디지털 홀로그램의 특성은 자연 영상과 매우 다른 특성을 갖기 때문에 2차원 영상에서 사용되던 변환 도구들을 디지털 홀로그램에 적용하는 것은 효율성이 매우 낮다. 이를 극복하기 위해 프레넬릿 변환이 제안되었는데 본 논문에서는 웨이블릿 기저함수에 유니터리 프레넬 변환을 적용하고 드뷔시 필터를 이용하여 프레넬릿 변환을 유도하였다. 또한 프레넬릿 변환을 디바이스 및 커널 코드를 이용하여 구현하여 동작 성능을 향상시키도록 하였다. 모든 해상도에 대해 한 화소당 소요되는 평균 시간을 기준으로 살펴보았을 때 디바이스 코드를 이용하여 병렬화 연산을 수행하면 (9,7)필터의 경우에는 평균 242배, (5,3)필터의 경우에는 평균 30배의 성능향상을 가져온다는 것을 확인하였다.

ABSTRACT

Service of digital hologram that has been recognized as a visual system for next generation requires various signal processing technologies. A transform is the most frequently used tool among signal processing techniques for 2-dimensional(D) and 3-D natural picture. A digital hologram has totally different property with a natural picture, so it is rarely efficient to apply transform tools used in 2-D image processing to a digital hologram. To overcome this a Fresnelet transform for a digital hologram has been proposed. We derive a Fresnelet transform by using the Daubechie's filter after applying an unitary Fresnel transform to a wavelet basis function. We also implement the transform as types of device and kernel code to improve operational performance. In consideration of the average time that is required for a pixel we can have observed the performance is improved up to 242 and 30 times for using the (9,7) and (5,3) filters in case of using device code.

키워드 : 프레넬릿, 웨이블릿, 프레넬, 디지털 홀로그램, 신호처리, 변환

Key word : Fresnelet, wavelet, Fresnel, digital hologram, signal processign, transform

Received 09 November 2016, **Revised** 02 December 2016, **Accepted** 20 December 2016

* **Corresponding Author** Dong-Wook Kim(E-mail:dwkim@kw.ac.kr, Tel:+82-2-940-5167)
Department of Electronic Materials Engineering, Kwangwoon University, Seoul 01897, Korea

Open Access https://doi.org/10.6109/jkiice.2017.21.4.820 print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/ by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. 서 론

홀로그래피란 1948년 헝가리 출신 영국 물리학자인 데니스 가보(Dennis Gabor)가 전자현미경의 분해능력을 향상시키기 위한 실험 중에 개발된 일종의 사진술로서, 홀로그램은 전체(완전한)라는 뜻의 “Holos”와 기록방법이라는 뜻의 “Gramme”가 합쳐져 만들어진 합성어로서 전체를 기록하는 방법이라는 의미를 가지고 있다. 1960년에 레이저 광선이 개발된 이후 비약적으로 발전을 한 3차원의 기록이 가능한 차세대 미디어 기술이다. 즉, 디지털 홀로그래피는 이미징 방식의 한 종류로써 디지털 홀로그램은 일반적으로 CCD 카메라를 이용하여 기록되거나 컴퓨터 생성 홀로그램(CGH, computer-generated hologram) 기법 등을 이용하여 생성되고, 수치적으로 복원된다. 홀로그램은 객체에 의해 반사 혹은 전파된 파와 참조파 사이의 간섭에 의한 결과이다. 객체 근처에서 진폭과 위상을 갖는 복소파의 디지털 복원은 회절 적분의 근사화인 프레넬 변환에 기초한다[1,2].

디지털 홀로그래피가 포함하는 분야는 너무 광범위하기 때문에 이들을 모두 설명하는 것은 매우 어려운 일이다. 최근들어 응용분야가 넓어짐에 따라, 영상 품질의 향상에 대한 요구가 증가되고 있다. 노이즈 억제, 고해상도의 복원 영상, 정밀한 파라미터 조절 그리고 빠르고 강건한 알고리즘들이 본질적인 이슈이다[3]. 이와 같이 디지털 홀로그래피의 신호처리를 위해서 필요한 도구 중의 가장 필수적인 것이 변환(transform) 도구이다. 기존의 연구들에서는 디지털 홀로그래피의 신호처리를 위해 2차원 영상처리에 사용되던 변환 도구들을 이용하였다[4-6]. 그러나 본질적으로 디지털 홀로그래피는 렌즈가 없는 처리 과정이기 때문에, 전체 영상 평면에 객체의 에지와 같은 고주파 성분을 확산하는 경향이 있다[7]. 구간적으로 부드러운 신호(부분적으로 2차원 영상이 갖는 저주파 영역, 이 영역에서 고압축이 가능)를 처리하기 위해 만들어진 이산 코사인 변환(DCT, discrete cosine transform)[7]이나 이산 웨이블릿 변환(DWT, discrete wavelet transform)[8]을 홀로그램에 직접적으로 적용할 때 최적화된 결과를 가져다 주지 못한다. 이를 극복하기 위해 디지털 홀로그램을 위한 맞춤형의 새로운 웨이블릿 기저함수를 설계하는 연구가 수행되었다[9,10]. 가우스-에르미트 함수들을 이용하여 회절문제를 분석적으로 해석할 수 있지만, 이러한 방법들은 웨이블릿 이론의 완비성(completeness)을 만족시키지 못하기 때문에 디지털 처리를 하는데 있어서 사용상 제약이 된다. 이러한 사실로부터 회절 문제에 적합한 기저함수를 찾아내었다. 이를 통해 변형된 웨이블릿 기저함수(basis function)를 만들기 위해 프레넬(Fresnel) 변환을 L_2 웨이블릿 기저함수에 적용함으로써 프레넬릿(Fresnelet) 함수를 제안하였다[9].

본 논문에서는 드뷔시의 필터 계수[11]를 가지고 [9]에서 제안된 프레넬릿 함수의 설계 방법을 이용하여 프레넬릿 변환을 구현하고자 한다. 또한 디지털 홀로그램의 실제적인 서비스를 위해서는 해상도가 매우 높기 때문에 처리 속도가 매우 중요한 이슈이다. 따라서 프레넬릿 변환을 CPU를 활용한 커널 코딩과 GPU를 활용한 디바이스 코딩 기법으로 구현하여 고속화시키고자 한다[12,13]. 본 논문은 다음과 같이 구성된다. 2장에서는 프레넬릿 변환을 위해 프레넬 변환의 수학적인 특성을 살펴본다. 이를 바탕으로 3장에서는 프레넬 변환의 특성을 웨이블릿 변환과 결합하여 프레넬릿 함수의 계수와 변환식을 정의한다. 4장에서는 구현 결과를 보이고 5장에서 결론을 맺는다.

Ⅱ. 프레넬 변환

자연 영상의 신호처리를 위해 이용되었던 주파수 변환도구들은 대부분 디지털 홀로그램의 부호화에서는 효율향상을 기대하기 어렵다. 이러한 이유에 의해 디지털 홀로그램의 공간 및 주파수적인 특성을 충분이 반영하면서 디지털 홀로그램을 해석하고 합성하기 위한 새로운 신호처리 도구에 대한 필요성이 대두되었다. 이러한 요구를 해결하기 위해 웨이블릿 변환과 프레넬 변환을 접목시킨 프레넬릿 변환기법이 제안되었다[9].

홀로그램은 물체로부터 회절된 파와 가간섭 기준파(coherent reference wave)로부터 생기는 간섭무늬(fringe pattern, interference pattern)인데, 이 과정은 광파(light wave)의 회절(diffraction)과 전달(translation)의 물리적 모델을 기본으로 한다. 그림 1과 같이 $E_0(\xi,\eta)=E(\xi,\eta,0)$를 물체면에서의 파면(wave plane)이라 하고, 이로부터 수직거리가 z인 평면에서의 파면을 $E(x,y,z)$이라 하면, 이에 대한 수학적 관계식은 식 (1)과 같이 키르히호프 적분으로 표현된다[3].

$$E(x,y,z) = E_0 * S(x,y,z) = \iint E_0(\xi,\eta) S(x-\xi, y-\eta, z) d\xi d\eta \quad (1)$$

여기서, $S(x,y,z) = -\frac{i\kappa}{2\pi z}\exp\left(i\kappa\sqrt{x^2+y^2+z^2}\right)$는 PSF (point-spread function)이며, λ는 주어진 광파의 파장이고 $\kappa = 2\pi/\lambda$이다.

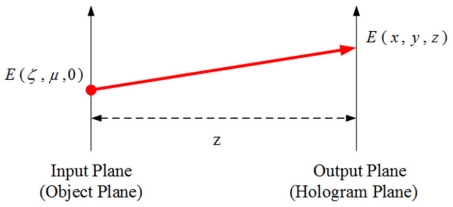


Fig. 1 Transmission process of light wave

충분히 큰 z, 즉, $\sqrt{(x-\xi)^2+(y-\eta)^2} \ll z$를 만족하는 프레넬 근사영역 내에 대하여 식 (2)와 같이 PSF를 근사하여 $E(x,y,z)$에 대해 식 (3)과 같이 프레넬 적분근사를 얻을 수 있다.

$$S(x,y,z) \approx -\frac{i\kappa}{2\pi z}\exp\left(i\kappa z + \frac{i\kappa(x^2+y^2)}{2z}\right) \quad (2)$$

$$E(x,y,z) \approx \frac{e^{i\kappa z}}{i\lambda z}\iint E_0(\xi,\eta)\exp\left(\frac{i\pi}{\lambda z}\left[(\xi-x)^2+(\eta-y)^2\right]\right)d\xi d\eta \quad (3)$$

여기서 적분커널을 $K_\tau(x,y)$를 $\exp\left(\frac{i\pi}{\tau^2}\left[x^2+y^2\right]\right)/\tau^2$라 두면, 식 (3)의 프레넬 근사는 간단히 식 (4)와 같이 된다. 여기서 $\tau = \sqrt{\lambda z}$이다. 식 (3)을 통해 두 개의 다른 깊이에서 파의 진폭과 위상은 2-D 프레넬 변환을 통하여 서로 연관된다는 것을 확인할 수 있다.

$$E(x,y,z) \approx -ie^{i\kappa z}E_0 * K_\tau(x,y) \quad (4)$$

그림 2는 샘플간격 $T = 41.8\,\mu\text{m}$인 200×200샘플로 표현된 입력 영상을 $\lambda = 630\,\text{nm}$, $z = 50\text{cm}$의 파라미터 값으로 프레넬 변환한 결과이다. 여기서 입력영상에서 막대의 길이는 $L = 2\,\text{mm}$이다. 이때 프레넬수는 $NF = (L/2)^2/\lambda z = 3.06$이다.

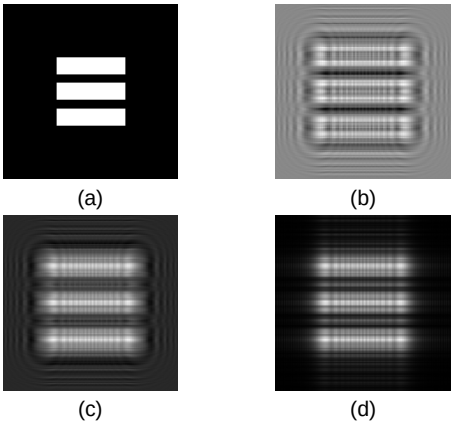


Fig. 2 Input image and Fresnel transform result ($\lambda = 630\,\text{nm}$, $z = 50\text{cm}$, $L = 2\,\text{mm}$, $T = 41.8\,\mu\text{m}$) (a) input, (b) real part, (c) imaginary part, (d) amplitude image

주어진 매개변수 $\tau > 0$에 대하여 함수 $f \in L_2(\mathbb{R}^2)$의 2차원 프레넬 변환은 식 (5)와 같이 K_τ를 커널함수로 하는 적분형 합성곱으로 정의할 수 있다.

$$\tilde{f}_\tau(x,y) = (f * K_\tau)(x,y) \quad (5)$$

여기서 커널함수 K_τ가 분리형이 되는데, 즉 $k_\tau(x) = \frac{1}{\tau}e^{i\pi(x/\tau)^2}$라 놓으면, 식 (6)과 같이 2차원 커널함수 K_τ를 1차원 함수의 텐서곱으로 표시 할 수 있게 된다.

$$K_\tau(x,y) = k_\tau(x)k_\tau(y) \quad (6)$$

따라서 1차원 프레넬 변환과 웨이블릿 변환을 이용하여 2차원 프레넬 변환과 웨이블릿 변환을 합성하여 프레넬릿을 정의할 수 있다. 1차원 프레넬 변환은 주어진 매개변수 $\tau > 0$와 함수 $f \in L_2(\mathbb{R})$에 대하여 식 (7)과 같이 커널함수 k_τ를 핵으로 하는 적분형 합성곱으로 정의된다.

$$\tilde{f}_\tau(x) = f * k_\tau(x) = \frac{1}{\tau}\int f(t)e^{i\pi[(t-x)/\tau]^2}dt \quad (7)$$

식 (7)에서 프레넬 커널 $k_\tau(x) = \frac{1}{\tau}e^{i\pi(x/\tau)^2}$의 푸리에 변환은 $\hat{k}_\tau(\omega) = e^{i\pi/4}e^{-i\pi(\tau\omega)^2}$이 되어, 프레넬 변환은 유니터리 변환이 됨을 알 수 있다. 따라서 주어진 함수에 대

해 완벽하게 재구성을 얻을 수 있다. 프레넬 변환의 유니터리 변환 특성과 선형적 성질을 이용하여 기존의 함수의 재구성과 합성하여 새로운 변환을 유도할 수 있다. 특히, $L_2(\mathbb{R})$ 공간의 직교기저 $\{\psi_l\}_{l\in\mathbb{Z}}$에 대하여 그의 프레넬 변환 $\{\Im_\tau(\psi_l)\}_{l\in\mathbb{Z}}$ (또는 $\left\{(\psi_l)_\tau^{\sim}\right\}_{l\in\mathbb{Z}}$)도 직교기저가 되므로, $\widetilde{f}_\tau$에 대해서도 마찬가지로 유사한 재구성을 얻을 수 있다.

Ⅲ. 프레넬릿 변환

본 장에서는 새로운 프레넬릿 기저를 만들기 위해, 웨이블릿 기저에 프레넬 변환을 적용한다. 영역 Ω의 차수는 $\Omega=\mathbb{R}$ 혹은 $\mathbb{R}^2$와 같이 임의로 두고, $L_2(\Omega)$의 일반적인 리에스 기저에 변환을 적용한다. 웨이블릿 기저는 스케일과 이동에 관한 두 개의 파라미터로 표현되는 함수열이므로 식 (8)과 같이 웨이블릿 함수가 정의된다[9].

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x-k),\ k,j\in\mathbb{N} \tag{8}$$

이에 대한 프레넬 변환은 이동과 팽창의 불변성으로 인해 식 (9)와 같이 나타낼 수 있다.

$$(\psi_{j,k})_{\tilde{\tau}}(x) = 2^{j/2}\widetilde{\psi}_{2^j\tau}(2^j x-k) \tag{9}$$

프레넬릿 변환은 주어진 직교 웨이블릿 ψ와 $\tau>0$에 대하여 식 (10)과 같이 정의할 수 있다.

$$\theta_{j,k}(x) = 2^{j/2}\widetilde{\psi}_{2^j\tau}(2^j x-k) \tag{10}$$

주어진 함수 f와 고정된 τ에 대해 식 (11)와 같이 직교 프레넬릿 기저를 얻을 수 있다.

$$f=\sum c_{j,k}\Theta_{j,k} \tag{11}$$

식 (11)에서 $\Theta_{j,k}(x)=(\Psi_{j,k})_{\tilde{\tau}}(x)$이고, $c_{j,k}$는 프레넬릿 계수에 해당하고, $c_{j,k}=\langle f,\Theta_{j,k}\rangle$이다. 웨이블릿 변환에서 다중해상도분석(mutiresolution analysis)은 데이터의 웨이블릿 재구성을 함에 있어 효율적인 알고리즘을 제공하는데, Fresnelet의 경우에도 다중해상도 분석을 구축할 수 있다.

프레넬 변환 및 웨이블릿 변환과 마찬가지로 분리 특성으로 인해 1차원 프레넬릿 변환은 2차원 프레넬릿 변환으로 확장될 수 있다. 이러한 특성으로부터 2차원 Fresnelet은 식 (12)에서 식 (15)와 같이 텐서곱으로 구축할 수 있다. 순서대로 각각 low-low 부대역, low-high 부대역, high-low 부대역, 그리고 high-high 부대역을 생성한다[10].

$$\gamma_\tau^{(ll)}=(\phi_{j,k})_{\tilde{\tau}}(x)(\phi_{j,k})_{\tilde{\tau}}(y) \tag{12}$$

$$\gamma_\tau^{(lh)}=(\phi_{j,k})_{\tilde{\tau}}(x)(\psi_{j,k})_{\tilde{\tau}}(y) \tag{13}$$

$$\gamma_\tau^{(ll)}=(\psi_{j,k})_{\tilde{\tau}}(x)(\phi_{j,k})_{\tilde{\tau}}(y) \tag{14}$$

$$\gamma_\tau^{(ll)}=(\psi_{j,k})_{\tilde{\tau}}(x)(\psi_{j,k})_{\tilde{\tau}}(y) \tag{15}$$

여기에서 ϕ는 스케일링 함수를 나타내고, ψ는 웨이블릿 함수를 나타낸다. 본 논문에서는 이 함수들에 드뷔시의 (9,7) 필터와 (5,3) 필터를 적용하여 구현한다[11]. 위의 기저함수를 함수 f에 적용하면 식 (16)에서 식 (19)와 같은 프레넬릿 계수를 얻을 수 있다[9,10].

$$f_{\tau,d}^{(ll)}=\langle f,\gamma_\tau^{(ll)}\rangle \tag{16}$$

$$f_{\tau,d}^{(lh)}=\langle f,\gamma_\tau^{(lh)}\rangle \tag{17}$$

$$f_{\tau,d}^{(hl)}=\langle f,\gamma_\tau^{(hl)}\rangle \tag{18}$$

$$f_{\tau,d}^{(hh)}=\langle f,\gamma_\tau^{(hh)}\rangle \tag{19}$$

그림 3은 드뷔시 필터를 이용하여 구현된 이산 프레넬릿 변환을 이용하여 입력영상을 3레벨만큼 Mallat 트리 방식으로 분해한 것이다. 여기서 $\lambda=632.8\,nm$이고 입력신호공간과 프레넬 변환공간 사이이의 거리는 $z=20\,cm$이다. 또한 입력신호의 샘플 간격은 $\Delta s=10\,\mu m$이다. 그림 3(a)는 원래의 객체에 해당하고, 그림 3(b)는 3레벨 부대역의 결과이다. 그림 3(c)는 부대역의 형태와 종류를 나타낸다. 그림 3(c)에서 L는 저주파 필터링을 의미하고 H는 고주파 필터링을 의미한다.

고전적인 웨이블릿 이론에서, 내장된 다해상도 공간들은 하나의 단일 함수의 확대와 이동을 통해서 생성된다. 프레넬 변환은 그러한 공간의 내장성(embeddedness)을 유지한다. 중요한 수정사항은 하나의 스케일에서 다음으로 생성 함수를 바꾸는 확대 관계로부터 비롯된다. 차이점은 변환 영역에서 각 스케일에 대해 하나의 생성 함수(generating function)만 존재한다는 것이다.

그림 4는 1레벨 프레넬릿 변환을 수행한 결과를 나타

낸다. 그림 4(a)는 진폭을 나타내고, 그림 4(b)와 (c)는 실수부와 허수부를 나타낸다. 프레넬 회절의 특성에 따라서 원래 객체의 정보가 거리에 따라 확산되는 현상을 확인할 수 있다. 이러한 특성을 통해서 구현한 프레넬릿 변환이 디지털 홀로그램을 위한 변환 도구로 적합하다는 것을 확인할 수 있다. 확산과 같은 광학적인 특성은 프레넬릿 역변환을 통해서 다시 수렴될 수 있어야 한다. 이에 대한 검증은 다음 장의 실험결과에서 디지털 홀로그램에 대한 결과로 보이도록 한다.

(a) (b) (c)

Fig. 3 Example of Fresnelet transform (a) input, (b) discrete Fresnelet transform result image, (c) subbands

(a) (b) (c)

Fig. 4 Subband images by Fresnelet transform (a) amplitude, (b) real, (c) imaginary part

Ⅳ. 실험결과

홀로그램 영상의 취득과정을 통하여 구축된 모델의 1차원 수학적 모델을 식 (20)과 같이 나타낼 수 있다[3].

$$\begin{aligned} I(x) &= \left|\tilde{f}_\tau(x) + R(x)\right|^2 \\ &= \left|\tilde{f}_\tau(x)\right|^2 + |R(x)|^2 + R^*(x)\tilde{f}_\tau(x) + R(x)\left(\tilde{f}_\tau(x)\right)^* \end{aligned} \tag{20}$$

여기서 f는 물체에서 반사된 파, R는 기준파 $R(x) = Ae^{i\omega x}$, 그리고 I는 CCD에 도달된 물체로부터 산란된 파 $\tilde{f}_\tau$와 기준파 R과 겹쳐진 간섭무늬이다. 위 식에서 주어진 홀로그램의 에너지는 주파수 영역에서 보면, 세 부분에 집중되어 있음을 알 수 있는데, $\left|\tilde{f}_\tau\right|^2 + |R|^2$는 0에, $R^*\tilde{f}_\tau$는 $-\omega$에, 그리고 $R\left(\tilde{f}_\tau\right)^*$는 ω에 그 에너지가 집중 된다. 기준파 R을 I에 적용하면 식 (21)과 같다.

$$RI = R\left(\left|\tilde{f}_\tau\right|^2 + |R|^2\right) + RR^*\tilde{f}_\tau + R\left(\tilde{f}_\tau\right)^* R \tag{21}$$

이때 RI 데이터는 I에서 $e^{i\omega x}$ 인수 만큼 변조된 것이다. 따라서 $RR^*\tilde{f}_\tau$, $R\left(\left|\tilde{f}_\tau\right|^2 + |R|^2\right)$, $R\left(\tilde{f}_\tau\right)^* R$은 각각 주파수 0, ω, 그리고 2ω에 집중되어 있다. 본 논문에서는 RI에 이산 프레넬릿 변환을 이용한 복원방법을 적용하여 세 항을 분리함과 동시에 f을 복원하는 기법을 확립하였다.

그림 5에 실험적인 디지털 홀래그래픽 데이터에 실제로 적용하여 다해상도 프레넬릿 기반 알고리즘을 검증하였다. 그림 5는 원영상에 표 1과 같은 값으로 프레넬 변환을 이용해 얻은 홀로그램 데이터 $I(x,y,z)$이다. 그림 5(a)의 회색 잡은과 같은 영상은 디지털 홀로그램을 나타내고, 이를 프레넬릿 함수를 이용하여 변환한 결과가 그림 5(b)이다. 역 프레넬릿 변환을 이용하여 디지털 홀로그램을 다시 복원한 이후에 프레넬 변환을 이용하여 디지털 홀로그램을 재생한 결과가 그림 5(c)이다. 그림 5(a)에 대해 프레넬 변환을 직접 적용한 결과와 그림 5(c)가 동일함을 확인하였기 때문에 구현한 프레넬릿 변환이 완전 복원(perfect reconstruction) 가능하다는 것을 확인하였다.

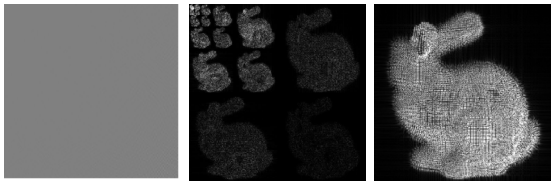
프레넬릿 변환은 많은 연산량을 요구하는 수학적인 처리과정이기 때문에 이를 고속화하기 위해 GPGPU (general purpose graphic processing unit)을 이용한 디바이스 프로그램과 CPU를 기반으로 하는 커널 프로그램으로 구현하여 성능을 비교하였다. 디바이스 프로그램을 위한 언어는 CUDA를 사용하였고[14], 커널 프로그램을 위한 언어는 C++을 이용하였다. 커널 프로그램만으로 구성된 코드, 커널 프로그램과 디바이스 프로그램을 혼용한 코드, 그리고 디바이스 프로그램을 위주로 구성된 코드로 구현하였다. 또한 각각 드뷔시의 (9,7)과 (5,3) 필터를 이용하여 구현하였다.

구현 결과를 표 2에서 표 7까지 나타냈다. 1,280×1,280, 1,920×1,920, 2,048×2,048, 4,096×4,096의 4가지 해상도를 갖는 디지털 홀로그램을 대상으로 프레넬릿 정변환(forward, FWD) 및 역변환(inverse, INV)하는데 소요되는 시간을 측정하였다. 또한 각 레벨(level, Lvl)

에 따른 시간도 각각 측정하였다. 표 2와 3은 커널 코드로만 구현된 경우의 결과이고, 표 4와 5는 커널 코드와 디바이스 코드를 혼용하여 구현한 결과이다. 표 6과 7은 거의 다바이스 코드만을 사용하고 최소한의 커널 코드를 사용하여 구현한 결과이다. 완전히 디바이스 코드만으로 프로그램을 만들 수는 없기 때문에 일부 커널 코드를 사용할 수 밖에 없다.

Table. 1 Experimental environment

#	Item	Specification
1	Wave length	$\lambda = 632.9nm$
2	Distance from an object to CCD plane	$z = 20cm$
3	Size of pixel pitch on CCD plane	$\Delta x = 12.395\mu m$
4	Size of pixel on CCD plane	$1,024 \times 1,024$



(a) (b) (c)

Fig. 5 Image example (a) digital hologram, (b) Fresnelet result (Level 4), (c) reconstructed object

표 8에는 표 2에서 표 7까지의 결과를 종합적으로 나타내었다. 표 2에서 7까지의 결과는 해상도와 레벨에 따라 필터 별로 얼마만큼의 연산 시간이 소요되는 것인지 나타낸 것이기 때문에 각 결과를 상호 비교하기 어렵다. 비교를 용이하게 하기 위해 프레넬릿 변환을 수행하는데 한 화소당 소요되는 시간을 구한 후에 모든 해상도에 대한 평균을 구하여 표 8에 정리하였다. 소요되는 시간의 단위는 [ns]이고, 커널 코드에 비해 성능이 몇 배 증가하는지를 나타냈다. 커널 코드과 비교할 때 디바이스 코드를 혼용하면 (9,7)필터의 경우에는 평균 50배, (5,3)필터의 경우에는 평균 7.5배의 성능향상을 가져온다. 디바이스 코드만 사용하면 (9,7)필터의 경우에는 평균 242배, (5,3)필터의 경우에는 평균 30배의 성능향상을 가져온다. 커널코드에 디바이스 코드를 혼용했을 경우에 비해서 디바이스 코드만 사용한 경우가 2배에서 2.6배만큼 성능이 더욱 향상되는 것으로 확인되었다. (5,3)코드의 경우는 부동소수점 연산이 없기 때문에 성능의 향상이 극적으로 이루어지지 않았다. 또한 레벨이 높아질

Table. 2 Calculation time of Fresnenet transform ((5,3) filter, kernel code)

Lvl	Dir.	Time for Resolution [ms]			
		1,280	1,920	2,048	4,096
1	FWD	531.62	797.43	14822.15	18061.59
	INV	442.76	664.12	16672.83	23176.31
2	FWD	572.28	858.41	16084.35	20876.46
	INV	501.04	751.57	18207.82	27418.32
3	FWD	597.50	896.26	17418.61	23555.71
	INV	542.89	814.34	19840.13	31790.969
4	FWD	625.36	938.04	18826.23	13634.47
	INV	587.59	881.39	21423.78	14902.69
5	FWD	654.54	981.80	20045.23	25355.38
	INV	632.75	949.12	22915.37	34127.63
6	FWD	679.96	1019.94	21905.90	28441.88
	INV	669.71	1004.57	25257.34	37205.18

Table. 3 Calculation time of Fresnenet transform ((9,7) filter, kernel code)

Lvl	Dir.	Time for Resolution [ms]			
		1,280	1,920	2,048	4,096
1	FWD	76.69	186.25	212.77	1011.29
	INV	90.87	217.52	247.59	1305.59
2	FWD	119.52	314.80	363.12	1801.26
	INV	165.25	407.39	467.93	2421.97
3	FWD	134.24	377.22	459.89	2552.06
	INV	203.70	559.79	641.12	3422.40
4	FWD	147.60	418.04	505.40	3195.36
	INV	222.55	617.37	759.06	4353.65
5	FWD	160.08	440.38	543.99	3635.53
	INV	238.30	653.88	828.81	5038.31
6	FWD	168.11	468.02	583.08	3926.12
	INV	250.89	679.06	890.97	5433.06

Table. 4 Calculation time of Fresnenet transform ((5,3) filter, kernel+device code)

Lvl	Dir.	Time for Resolution [ms]			
		1,280	1,920	2,048	4,096
1	FWD	76.69	186.25	212.77	1011.29
	INV	90.87	217.52	247.59	1305.59
2	FWD	119.52	314.80	363.12	1801.26
	INV	165.25	407.39	467.93	2421.97
3	FWD	134.24	377.22	459.89	2552.06
	INV	203.70	559.79	641.12	3422.40
4	FWD	147.60	418.04	505.40	3195.36
	INV	222.55	617.37	759.06	4353.65
5	FWD	160.08	440.38	543.99	3635.53
	INV	238.30	653.88	828.81	5038.31
6	FWD	168.11	468.02	583.08	3926.12
	INV	250.89	679.06	890.97	5433.06

Table. 5 Calculation time of Fresnenet transform ((9,7) filter, kernel+device code)

Lvl	Dir.	Time for Resolution [ms]			
		1,280	1,920	2,048	4,096
1	FWD	75.35	176.98	221.56	1045.34
	INV	86.84	206.07	320.42	1576.77
2	FWD	115.74	296.24	376.04	1851.15
	INV	144.80	371.87	584.12	3049.02
3	FWD	145.69	359.29	521.80	2569.22
	INV	186.89	459.64	827.93	4496.29
4	FWD	174.07	421.65	611.99	3180.77
	INV	227.67	534.23	960.02	5635.46
5	FWD	202.87	487.78	703.32	3747.76
	INV	265.06	610.29	1104.56	6610.4
6	FWD	229.12	549.82	809.63	4120.97
	INV	298.30	669.31	1245.29	7152.18

Table. 6 Calculation time of Fresnenet transform ((9,7) filter, device code)

Lvl	Dir.	Time for Resolution [ms]			
		1,280	1,920	2,048	4,096
1	FWD	22.65	50.33	68.34	141.58
	INV	21.52	46.99	63.61	137.59
2	FWD	29.35	66.07	93.62	264.03
	INV	27.74	61.68	88.56	254.25
3	FWD	41.66	85.07	116.05	362.60
	INV	40.10	81.17	110.60	350.15
4	FWD	66.18	121.67	154.56	448.47
	INV	64.91	117.83	150.64	429.51
5	FWD	116.30	193.91	236.85	604.87
	INV	115.49	190.77	235.88	588.30
6	FWD	217.80	350.91	421.21	945.91
	INV	217.05	340.64	436.11	933.31

Table. 7 Calculation time of Fresnenet transform ((5,3) filter, device code)

Lvl	Dir.	Time for Resolution [ms]			
		1,280	1,920	2,048	4,096
1	FWD	22.33	53.12	69.62	142.85
	INV	31.35	49.14	95.48	132.04
2	FWD	31.35	71.05	95.48	270.48
	INV	30.76	68.15	90.81	265.44
3	FWD	47.63	94.95	120.94	377.07
	INV	47.32	90.90	116.56	365.99
4	FWD	76.08	136.40	164.56	474.17
	INV	75.79	131.37	160.25	458.47
5	FWD	129.71	216.36	259.07	649.68
	INV	128.19	213.46	255.02	641.19
6	FWD	246.07	387.58	481.64	1032.42
	INV	243.90	382.90	476.85	1013.20

수록 성능이 향상되는 정도가 낮아진다. 이는 레벨을 거치면서 중간 결과들을 디바이스로부터 CPU로 옮기도록 프로그램이 구성되어 있기 때문에 주메모리의 접근이 늘어남에 따라 성능이 저하되었기 때문이다. 디바이스 코드에서 가장 많은 시간을 소요하면서 병목현상을 발생시키는 것이 CPU에서 GPGPU의 전역 메모리(global memory)로 데이터를 복사하는 과정이다[14].

Ⅴ. 결 론

디지털 홀로그램의 공간과 주파수적인 특성을 충분이 반영하여 디지털 홀로그램을 해석하고 합성하기 위한 새로운 신호처리 도구로 프레넬릿 변환이 제안되었는데, 본 논문에서는 웨이블릿 기저함수에 유니터리 프레넬 변환을 적용하고 드뷔시 필터를 이용하여 새로운 프레넬릿 변환을 유도하였다. 구현한 프레넬릿 변환은 디지털 홀로그램을 효율적으로 분해하는 것을 확인할 수 있었고, 완전 복원이 가능하다는 것을 검증하였다. 또한 프레넬릿 변환을 디바이스 코드와 커널 코드를 이용하여 구현하였다. 모든 해상도에 대해 한 화소당 소요되는 평균 시간을 기준으로 살펴보았을 때 디바이스 코드를 이용하여 병렬화 연산을 수행하면 (9,7)필터의 경우에는 평균 242배, (5,3)필터의 경우에는 평균 30배의 성능향상을 가져온다는 것을 확인하였다. 추후에 본 연구팀은 다양한 웨이블릿 함수들을 이용하여 다양한 프레넬릿 변환을 구현하고자 하고, 이를 통해 최적화된 프레넬릿 함수를 찾기 위한 연구를 지속할 예정이다.

Table. 8 Performance comparison

Lvl	Filter Tap	Kernel		Kernel+Device		Device	
		[ns]	%	[ns]	%	[ns]	%
1	(9,7)	5479.07	1	61.05	89.75	12.68	432.03
	(5,3)	700.32	1	57.46	12.19	14.53	48.19
2	(9,7)	6073.16	1	107.66	56.41	17.98	337.82
	(5,3)	764.45	1	102.45	7.462	19.00	40.23
3	(9,7)	6671.08	1	146.01	45.69	23.94	278.66
	(5,3)	868.01	1	134.88	6.435	26.16	33.18
4	(9,7)	6265.59	1	175.60	35.68	33.76	185.59
	(5,3)	974.70	1	157.28	6.20	37.30	26.14
5	(9,7)	7548.78	1	203.99	37.01	53.71	140.56
	(5,3)	1092.66	1	173.04	6.32	59.19	18.46
6	(9,7)	8265.21	1	226.81	36.44	96.18	85.94
	(5,3)	1216.41	1	184.53	6.59	107.31	11.34

ACKNOWLEDGMENTS

The present Research has been conducted by the Research Grant of Kwangwoon University in 2017.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2014R1A2A1A11052433).

REFERENCES

[1] J. W. Goodman and R. W. Lawrence, "Digital image formation from electronically detected holograms," *Applied Physics Letters*, vol. 11, no. 3, pp. 77-79, Sept. 1967.

[2] D. Gabor, "A new microscopic principle," *Nature*, vol. 161, no. 4098, pp. 777-778, May 1948.

[3] J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, 3rd ed. pp. 377-393, New York, 2005.

[4] Y. H. Seo, Y. H. Lee, J. M. Koo, W. Y. Kim, J. S. Yoo, and D. W. Kim, "Digital holographic video service system for natural color scene," *Optical Engineering*, vol. 52 no. 11, pp. 1-8, Nov. 2013.

[5] Y. H. Seo, H. J. Choi and D. W. Kim, "Digital hologram encryption using discrete wavelet packet transform," *Optics Communications*, vol. 282, no. 3, pp. 367-377, Jan. 2009.

[6] H. J. Choi, Y. H. Seo, J. S. Yoo, and D. W. Kim, "Digital watermarking technique for holography interference patterns in a transform domain," *Optics and Lasers in Engineering*, vol. 46, no. 4, pp. 343-348, Apr. 2008.

[7] K. Ramamohan Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, 1st ed. Massachusetts, Jun. 2014.

[8] R. M. Rao and A. S. Bopardikar, *Wavelet Transforms, Introduction to Theory and Application*, Prentice Hall, Har/Dskt ed. New Jersey, Aug. 1998.

[9] M. Liebling, T. Blu and M. Unser, "Fresnelets: new multiresolution wavelet bases for digital holography," *IEEE Transaction Image Process*, vol. 12, no. 1, pp. 29-43, Jun. 2003.

[10] N. Muhammad, N. Bibi, Z. Mahmood and D. G. Kim, "Blind data hiding technique using the Fresnelet transform," *SpringerPlus*, vol. 4, no.1, pp. 1-15, Dec. 2015.

[11] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1st ed. Philadelphia, Jun. 1992.

[12] J. S. Song, J. S. Park, H. H. Park, and J. I. Park, "Real-time generation of high-definition resolution digital holograms by using multiple graphic processing units," *Optical Engineering*, vol. 52, no. 1, pp. 015803-015803, Jan. 2013.

[13] Y. H. Seo, Y. H. Lee, J. S. Yoo, and D. W. Kim, "High-performance Computer-generated Hologram by Optimized Implementation of parallel GPGPUs," *Journal of the Optical Society of Korea*, vol. 18, no. 6, pp. 698-705, Dec. 2014.

[14] S. Cook, *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of Gpu Computing)*, Morgan Kaufmann, 1st ed. Massachusetts, Nov. 2012.

서영호(Young-Ho Seo)

1999년 2월 : 광운대학교 전자재료공학과 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 홀로그램, SoC 설계

이윤혁(Yoon-Hyuk Lee)

2012년2월 : 광운대학교 전자재료 공학과 졸업(공학사)
2014년2월 : 광운대학교 일반대학원 졸업(공학석사)
2014년3월 ~ 현재 : 광운대학교 일반대학원 박사과정
※관심분야: 디지털 홀로그램, SoC 설계

김동욱 (Dong-Wook Kim)

1983년 2월 한양대학교 전자공학과 졸업(공학사)
1985년 2월 한양대학교 공학석사
1991년 9월 Georgia공과대학 전기공학과(공학박사)
1992년 3월 ~ 현재 광운대학교 전자재료공학과 정교수
※관심분야 : 3D 영상처리, 디지털 홀로그램, 디지털 VLSI Testability, VLSI CAD, DSP설계