

3차원 동영상의 시·공간적 정보 중복성을 이용한 효과적인 3차원 비디오 홀로그램의 생성

김 동 욱*, 구 정 식*, 김 승 철**, 김 은 수^o

Efficient Generation of 3-D Video Holograms Using Temporal-Spatial Redundancy of 3-D Moving Images

Dong-Wook Kim^{*}, Jung-sik Koo^{*}, Seung-Cheol Kim^{**}, Eun-Soo Kim^o

요 약

본 논문에서는 3차원 동영상의 시·공간적 정보 중복성 (TSR: temporal-spatial redundancy)과 노블 룩업테이블 (N-LUT: novel look-up table)를 이용한 효과적인 3차원 동영상 홀로그램 생성기법인 TSR-N-LUT을 제시하였다. 제안된 기법에서는 먼저, DPCM (differential pulse code modulation) 알고리즘을 이용하여 3차원 동영상 프레임 간의 시간적 중복 데이터를 제거한 다음, 잔여 영상에 대해서도 각 프레임별로 영상라인 간 중복 데이터를 제거하게 된다. 따라서, 시·공간적인 중복 데이터가 제거된 물체영상에 N-LUT 기법을 적용함으로써 효과적으로 3차원 비디오 홀로그램을 합성하게 된다. 실험결과 제안된 TSR-N-LUT 기법은 기존 방법에 비해 평균적으로 홀로그램 계산을 위한 물체 포인트 수가 23.72% 감소함과 동시에 홀로그램 생성속도 역시 19.55% 개선됨을 보임으로써 제안된 기법의 실제 응용 가능성을 제시하였다.

Key Words : CGH, N-LUT, TR-N-LUT, LR-N-LUT, DPCM

ABSTRACT

In this paper, a new method to efficiently generate the 3-D(three-dimensional) video holograms for 3-D moving scenes, which is called here the TSR-N-LUT method, is proposed by the combined use of temporal-spatial redundancy(TSR) of 3-D video images and novel look-up table(N-LUT) technique. That is, in the proposed scheme, with the differential pulse code modulation (DPCM) algorithm, temporally redundancy redundant data in the inter-frame of a 3-D video images are removed between the frames, and then inter-line redundant data in the inter-frame of 3-D video images are also removed by using the DPCM method between the lines. Experimental results show that the proposed method could reduced the number of calculated object points and the calculation time of one object point by 23.72% and 19.55%, respectively on the average compared to the conventional method. Good experimental results with 3-D test moving pictures finally confirmed the feasibility of the proposed method to the fast generation of CGH patterns of the 3-D video images.

I. 서 론

홀로그래피(holography) 기술은 원리적으로 완전한 3차원 입체영상 기술로 개발 초부터 3차원 디스

※ 본 연구는 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임. (NO. 2012-0009224)

• 주저자: 광운대학교 전자공학과 홀로 디지로그 휴먼 미디어 연구센터, analog_mind@naver.com, 준회원

◦ 교신저자: 광운대학교 전자공학과 홀로 디지로그 휴먼 미디어 연구센터, eskim@kw.ac.kr, 종신회원

* 광운대학교 전자공학과 홀로 디지로그 휴먼 미디어 연구센터, holocube@naver.com, 정회원

** 광운대학교 전자공학과 홀로 디지로그 휴먼 미디어 연구센터, sckim@kw.ac.kr, 정회원

논문번호: KICS2012-09-453, 접수일자: 2012년 9월 24일, 최종논문접수일자: 2012년 10월 14일

플레이 기술로 활용하기 위한 많은 연구가 이루어졌으나 여러 가지 현실적인 문제로 그 응용이 실질적으로 제한되어 왔다¹¹. 즉, 기존의 광 홀로그램(optical hologram)은 가 간섭성 광인 레이저(laser) 광을 사용하여 만들어지기 때문에 실험공간이 암실로 제한되고, 진폭과 위상정보를 저장하기 위한 과정에서 작은 움직임에도 간섭무늬가 파괴될 수 있기 때문에 매우 안정적인 광학 시스템이 요구된다. 이러한 문제를 해결하는 새로운 접근방법으로 컴퓨터를 이용하여 홀로그램 패턴을 생성하는 컴퓨터형성 홀로그램(CGH: computer-generated hologram) 기법이 제안되었다¹²⁻⁴¹.

일반적으로, 임의 물체에 대한 디지털 홀로그램 패턴을 계산할 때 빛의 회절을 계산하는 광선추적(ray-tracing) 방식이 주로 사용되어 왔다. 이 방식에서는 임의의 물체영상을 점(object point)들의 집합으로 보고 각각의 물체 점들에 대한 홀로그램 패턴을 모두 계산하여 합산하게 된다. 따라서 이 방법은 과도한 계산량이 요구되기 때문에 복잡한 물체에 대한 실시간 홀로그램패턴 합성을 어렵게 한다⁵¹. 이러한 문제점을 극복하기 위하여 룩업테이블(LUT: look-up table)을 이용한 새로운 디지털 홀로그램 계산 방법이 제안되었다⁶¹. 이 방법은 모든 가능한 물체 포인트에 대한 요소프린지패턴(EFP: elemental fringe pattern)을 미리 계산하여 데이터베이스로 저장하게 된다. 따라서 임의 물체에 대한 홀로그램 계산은 그 물체를 구성하고 있는 각 포인트에 대한 EFP들을 이미 계산하여 저장해 놓은 데이터베이스에서 단순히 불러내어 합산함으로써 구하게 된다. 이러한 룩업테이블 방식은 광선추적 방식에 비해 계산량이 크게 감소하여 디지털 홀로그램의 고속 생성이 가능하다. 하지만 이 방법은 물체영역이 커질수록 필요로 하는 EFP의 개수가 크게 늘어나게 되고, 결국은 룩업테이블의 메모리 용량이 방대해지는 단점을 가지고 있다.

최근, 기존의 LUT 방식의 장점인 고속의 연산속도는 그대로 유지를 하면서도 기존 LUT 방식의 단점인 방대한 메모리 공간을 줄일 수 있는 새로운 접근방법인 노블 룩업테이블(N-LUT: novel look-up table) 방식이 제안되었다⁷¹. N-LUT 방법에서는 각 물체평면별로 대표요소프린지패턴(PFP: principle fringe pattern) 한 개만을 미리 만들어 룩업테이블에 저장하고, 수평 방향으로 각각 이동시켜 평면상의 모든 물체 점들을 표현하게 된다.

따라서 N-LUT 방식에서는 물체의 깊이 방향에

대한 PFP 만을 저장하게 되고, 물체의 한 깊이 방향이 결정되면 그 면에 존재하는 모든 물체 포인트들의 요소프린지패턴들은 사전에 저장된 그 깊이의 대표요소 프린지패턴을 각 물체 포인트까지 이동시켜 프린지 패턴을 계산하여 그 깊이 평면에서의 홀로그램 패턴을 계산하게 된다. 같은 방법으로 모든 물체 깊이 평면에서 각각의 홀로그램을 계산하여 합산함으로써 물체 전체에 대한 홀로그램 패턴을 계산하게 된다. 따라서 기존의 LUT 방식의 고속 생성 특성은 유지하면서 메모리 용량을 획기적으로 줄일 수 있게 되었다.

한편, N-LUT 방식에서는 물체 포인트 단위로 홀로그램 생성을 수행하게 된다. 따라서 계산할 홀로그램의 정보가 늘어나게 되면 홀로그램 생성 시간 또한 선형적으로 증가하게 된다.

이러한 문제점을 해결하기 위하여 최근, 3차원 영상에 대하여 중복성 데이터를 추출하여 계산할 홀로그램 데이터를 줄여 생성시간을 줄일 수 있는 방법들이 제안되었다⁸⁻¹²¹. 즉, 3차원 정지영상에 대해서는 인접한 픽셀들 간의 정보를 비교하여 같은 값을 갖게 되면 그룹화 하여 한 번에 연산을 함으로써 홀로그램 생성시간을 단축할 수 있는 공간적 중복성(SR: spatial redundancy) 기반의 새로운 N-LUT 방식인 SR-N-LUT 기법이 제시되었다⁹¹.

한편, 3차원 동영상에 대해서는 프레임간의 시간적 중복성 데이터를 추출한 후 이전 프레임에 비해 변화가 생긴 부분에 대한 보정만을 해줌으로써 홀로그램 생성시간을 단축할 수 있는 시간적 중복성(TR: temporal redundancy) 기반의 새로운 N-LUT 방식인 TR-N-LUT 기법이 제시되었다¹⁰¹. 그러나 3차원 동영상의 경우 프레임간의 시간적 중복 데이터가 제거된 이후의 영상 역시, 3차원 정지영상의 시퀀스이기 때문에 각 프레임 영상 내에서의 공간적 중복내의 인터라인 정보까지도 제거하게 되면 보다 효과적인 3차원 동영상 홀로그램의 합성이 가능하게 된다.

따라서 본 논문에서는 3차원 동영상의 시·공간적 중복성(TSR: temporal-spatial redundancy)을 동시에 고려한 새로운 형태의 3차원 비디오 홀로그램 생성기법인 TSR-N-LUT 기법을 제시하고자 한다.

즉, 제안된 기법에서는 DPCM (differential pulse code modulation)을 이용하여 3차원 동영상 프레임간의 시간적 중복 데이터를 제거하고, 남은 영상에 대해서도 각 프레임별로 공간적 중복 내의 인터라인 데이터를 추가로 제거하게 된다¹¹¹. 따라서 시·

공간적 중복 데이터가 모두 제거된 입력영상에 N-LUT 알고리즘을 적용함으로써 최종적으로 3차원 동영상 홀로그램을 합성하게 된다. 3차원 테스트 영상을 사용한 실험을 통해 기존 방법에 비해 제안된 TSR-N-LUT 기법에서 홀로그램패턴 계산을 위한 물체 포인트 수가 크게 감소하고 결과적으로 고속의 3차원 동영상 홀로그램 패턴의 합성이 가능함을 보임으로써 제안된 기법의 실제 응용 가능성을 제시하고자 한다.

II. 기존의 N-LUT 기법

2.1. 기존의 LUT 방식

디지털 홀로그램 패턴은 픽셀들의 좌표 값과 세기 값에 의해 컴퓨터 계산으로 생성된다. 즉, 디지털 홀로그램 패턴을 생성하는 기하학적 구조는 [Fig. 1]과 같으며, [Fig. 1]에서 홀로그램은 $z=0$ 인 x - y 평면에 위치하고, 물체의 p 번째 점은 (x_p, y_p, z_p) 에 위치하고 a_p 의 세기값과 ϕ_p 의 위상값을 갖는다고 가정한다.

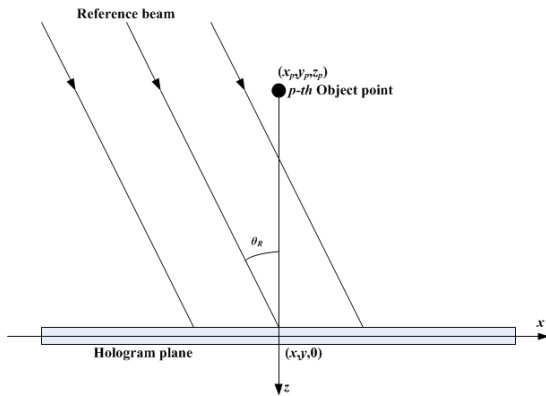


Fig. 1. Geometry for generation of the Fresnel holograms

이 때, 홀로그램에서 복소 진폭 $O(x, y)$ 는 식 (1)에서 나타난 것처럼 물체파의 중첩으로 구할 수 있다.

$$O(x, y) = \sum_{p=1}^N \frac{a_p}{r_p} \exp[j(kr_p + \phi_p)] \quad (1)$$

여기서 N , a_p 는 물체를 구성하는 점들의 수와 물체파의 세기를 각각 나타내고, 파수(wave number) 벡터 k 는 $k = 2\pi/\lambda$ 로 정의되며 λ 는 자유공간상에서의 빛의 파장을 각각 나타낸다. 또한, 평면파인 기준파의 복소진폭 $R(x, y)$ 는 식 (2)과 같이 주어진다.

$$R(x, y) = a_R \exp[j(-kx \sin \theta_R)] \quad (2)$$

여기서 a_R 과 θ_R 은 기준파의 세기와 입사각을 각각 나타낸다. 홀로그램 면에서의 전체적인 격자 세기는 식 (3)과 같이 물체파 $O(x, y)$ 와 기준파 $R(x, y)$ 사이의 간섭패턴으로 주어진다.

$$I(x, y) = |O(x, y) + R(x, y)|^2 = |R(x, y)|^2 + |O(x, y)|^2 + 2|R(x, y)||O(x, y)|\cos[kr_p + kx \sin \theta_R + \phi_p] \quad (3)$$

식 (3)에서 첫 번째와 두 번째 항은 물체파와 기준파의 세기를 각각 나타내며, 세 번째 항은 홀로그램 정보를 부분적으로 포함하고 있는 물체파와 기준파 사이의 간섭 패턴을 의미한다. 식 (4)에서 홀로그램 정보는 오직 세 번째 항에만 포함되어 있기 때문에 홀로그램 패턴 $I(x, y)$ 는 최종적으로 아래와 같이 표현될 수 있다.

$$I(x, y) = 2 \sum_{p=1}^N \frac{a_p}{r_p} \cos(kr_p + kx \sin \theta_R + \phi_p) \quad (4)$$

디지털 홀로그램의 연산속도를 향상시키기 위한 새로운 접근방법으로 룩업테이블(LUT) 방식이 M. Lucent에 의해 제안되었다⁶⁾. 이 방식은 가능한 물체영상 공간 내의 모든 점 (x_p, y_p, z_p) 에 대하여 기준 세기의 요소프린지패턴(EFP) $I(x, y)$ 를 미리 계산하여 테이블로 만들어 놓은 후에, 물체에 따라 필요한 포인트에 대한 EFP를 불러내어 사용하는 방식이다.

일반적으로 실세계의 공간은 연속적이다. 따라서 실세계의 공간은 무한한 점을 갖게 되고 룩업테이블을 이용해 홀로그램을 만들기 위해서는 무한개의 EFP가 필요하게 된다. 하지만 인간의 시각 시스템의 능력은 제한되어 있기 때문에 물체 점들을 이산화 하여 홀로그램을 합성하여도 물체 영상의 해상도 저하 없이 영상을 재생할 수 있게 된다. 이때 이산화(discretization) 정도는 사람의 눈에 인지되지 않을 정도로 충분히 작아 연속적인 두 개의 점이 떨어져 있지 않고 연속적인 점으로 인식될 수 있어야 한다. 예를 들면, 인간은 3 milliradian의 간격을 갖는 두 점을 하나의 점으로 인식하게 된다. 따라서 500 mm의 거리에서 물체영상을 바라보는 경우는 $500 \text{ mm} \times 0.003 = 150 \text{ microns}$ 이하의 간격의

점을 하나의 점으로 인식하게 됨으로 본 논문에서는 수직, 수평의 이산화 정도를 150 micron으로 사용하였다.

LUT 방식의 EFP는 식 (1)로부터 생성해 낼 수 있다. 즉, 기준 세기를 갖는 EFP $T(x, y; x_p, y_p, z_p)$ 는 다음과 같이 표현된다.

$$T(x, y, x_p, y_p, z_p) \equiv \frac{1}{r_p} \cos [kr_p + kx \sin \theta_R + \phi_p] \quad (5)$$

여기서, r_p 는 p 번째 점과 홀로그램 평면 $(x, y, 0)$ 사이의 거리로 식 (6)과 같이 주어진다.

$$r_p = \sqrt{(x - x_p)^2 + (y - y_p)^2 + z_p^2} \quad (6)$$

이 방식에서는 홀로그램을 계산할 때 식 (4)와 같이 각 점에 대한 프린지 패턴을 필요할 때마다 계산하는 것이 아니라 미리 만들어놓은 각 점 (x_p, y_p, z_p) 에 대한 프린지 패턴의 집합인 룩업테이블을 이용하여 계산을 하게 된다. 따라서 LUT 방식에서 홀로그램 정보 $I(x, y)$ 는 최종적으로 식 (7)과 같이 주어진다.

$$I(x, y) = \sum_{p=1}^N a_p T(x, y; x_p, y_p, z_p) \quad (7)$$

위의 식에서 N 은 물체 포인트의 개수를 나타낸다.

2.2. 기존의 N-LUT 방식

최근, 기존의 LUT 방식의 장점인 고속의 연산속도는 그대로 유지하면서 기존 LUT 방식의 단점인 방대한 메모리 용량을 크게 줄일 수 있는 N-LUT 방식이 제안되었다⁷⁾. 이 방식에서는 3차원 물체를 각각의 깊이에 해당하는 평면들의 집합으로 모델링하고, 각 평면은 자체 발광하는 물체 점들의 집합으로 간주하게 된다. 그리고 각 평면의 중앙에 있는 물체 점에 해당하는 대표요소프린지패턴인 PFP를 미리 만들어 룩업테이블에 저장하고, 이를 불러와 x, y 방향으로 각각 이동시켜 평면상의 모든 점들을 표현하게 된다.

즉, 각각의 PFP $T(x, y; z_p)$ 는 임의의 깊이 z_p 에 대한 기준 세기의 프레넬 존 플레이트(Fresnel zone plate)가 되고 다음과 같이 주어진다.

$$T(x, y; z_p) \equiv \frac{1}{r_p} \cos [kr_p + kx \sin \theta_R + \phi_p] \quad (8)$$

여기서 r_p 는 p 번째 점과 홀로그램 평면 $(x, y, 0)$ 사이의 거리로 식 (6)과 같이 주어진다.

따라서 N-LUT 방식에서는 물체의 깊이 방향에 대한 PFP만을 계산하여 저장하게 되고, 물체의 한 깊이 방향이 결정 되면 그 면에 존재하는 물체 포인트들의 요소프린지패턴들은 사전에 저장된 그 깊이의 대표요소프린지패턴을 각 물체 포인트까지 이동시켜 프린지 패턴을 계산하여 그 깊이 평면에서의 홀로그램 패턴을 계산하게 된다. 같은 방법으로 모든 물체 깊이 평면에서 각각의 홀로그램을 계산하여 합산함으로써 물체 전체에 대한 홀로그램 패턴을 계산하게 된다.

따라서 N-LUT 방식에서 홀로그램 정보 $I(x, y)$ 는 다음과 같이 표현할 수 있다.

$$I(x, y) = \sum_{p=1}^N a_p T(x - x_p, y - y_p; z_p) \quad (9)$$

2.3. 기존의 3차원 동영상의 시간적 중복성 기반의 홀로그램 생성(TR-N-LUT)

일반적인 3차원 영상을 보면 일반 2차원 영상과 마찬가지로 초당 30장의 정지영상으로 이루어져 있다. 그러므로 1/30초 단위로 정지영상을 촬영하여 연결한 것이라고 볼 수 있다. 화면전환이 존재하지 않는 한 1/30초는 연속된 영상간의 상관성이 높을 만큼 충분히 짧은 시간이다. 즉, 동영상은 인접한 영상들이 매우 유사하다.

이는 연속된 영상 간에 중복된 정보들이 매우 많다는 것을 의미하고 이러한 중복 정보를 시간적 중복성(temporal redundancy) 정보라고 하고, 최근 이러한 시간적 중복성 정보와 N-LUT를 이용하여 3차원 동영상 홀로그램을 효율적으로 생성하는 방법인 TR-N-LUT 기법이 제안되었다¹⁰⁾.

이 기법에서는 이전 프레임과 현재 프레임간의 차이를 추출한 후 이전 프레임의 홀로그램에서 변화된 부분만의 홀로그램을 생성하여 보상하여 주어 계산 속도를 향상시켰다.

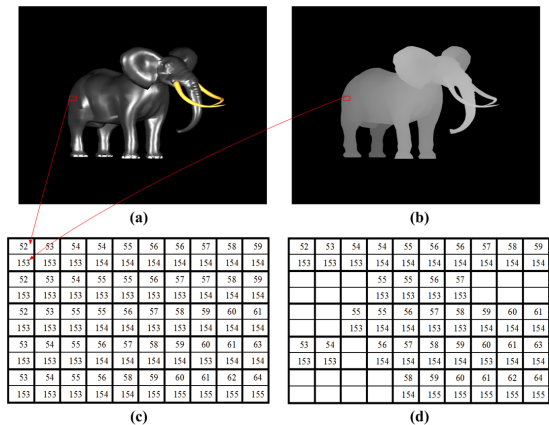
2.4. 기존의 3차원 영상의 인터라인 중복성 기반의 홀로그램 생성(LR-N-LUT)

3차원 동영상의 프레임간의 유사성과 같이 3차원

영상의 인접한 라인들의 간에도 유사성을 갖게 된다. 즉, 3차원 영상의 인접한 라인의 같은 위치에 있는 점들은 유사한 값을 갖게 되는데 이러한 라인 간 중복성 정보를 인터라인 중복성(In-line redundancy) 정보라고 하고, 최근 이러한 인터라인 중복성 정보와 N-LUT를 이용하여 3차원 동영상 홀로그래ム을 효율적으로 생성하는 방법인 LR-N-LUT 기법이 제안되었다^[11].

[Fig. 2(a)]와 [Fig. 2(b)]를 보면 밝기와 깊이 정보를 나타낸다. [Fig. 2(c)] 두 번째 라인을 보면 1~3번, 8~10번 포인트의 경우 밝기 와 깊이 정보가 첫 번째 라인과 같은 것을 볼 수 있다.

세 번째 라인의 경우는 1, 2번 포인트만 두 번째 라인과 같은 값을 갖고 나머지는 모두 다른 값을 갖는 것을 볼 수 있다. 마찬가지로 다섯 번째 라인의 경우 1~4번째 포인트는 같은 값을 갖지만 나머지 포인트는 다른 값을 갖는 것을 볼 수 있다. [Fig. 2(d)]는 이로부터 이전 라인과의 차이를 추출한 맵을 나타낸다. 즉, 이전 라인과 같은 값을 갖는 경우 이전 라인의 정보를 이용하고 다른 값을 갖는 경우 현재 라인의 정보를 이용하게 된다. 따라서 홀로그래ム 생성을 위해서는 이전 라인에 대한 홀로그래ム을 이산화 정도에 따라 이동시킨 후 라인간의 차이만을 보상해 줌으로써 현재 라인의 홀로그래ム을 생성하게 된다. 그리고 이를 마지막 라인까지 수행하여 전체 영상에 대한 홀로그래ム을 생성하여 계산 속도를 향상시켰다.



(a) Intensity image (b) depth image (c) pixel intensity and depth values (d) Interline redundancy map of the 3-D image
 Fig. 2. Intensity and depth values of a 3-D image:

III. 제안된 TSR-N-LUT 기법

기존의 LUT 방식에서는 가능한 물체영상의 모든 포인트에 대해 미리 계산 해놓은 EFP를 이용함으로써 획기적인 홀로그래ム 생성 속도의 증가를 가져왔다. 그러나 이 방식의 가장 큰 단점은 미리 계산해 놓은 EFP의 데이터양이 워낙 많고 따라서 이를 저장할 LUT의 메모리 또한 엄청나게 증가하게 된다는 것이다. 예를 들어, 기존의 LUT 방식에서 물체 공간이 100(가로) × 100(세로) × 100(깊이) 이고, 각 요소프린지패턴의 용량이 1MB라고 가정하면 전체 LUT의 메모리 용량은 1TB(1MB × 100 × 100 × 100) 에 까지 이르게 된다.

이러한 문제점을 해결하기 위해 제안된 N-LUT 방식에서는 고속의 홀로그래ム 계산 속도는 그대로 유지하면서 필요로 하는 메모리 공간은 현저히 줄어들게 된다. 하지만 계산 속도가 현저히 줄어든다 하더라도 동영상의 경우 초당 30프레임의 영상이 존재하기 때문에 계산해야 할 전체 물체 데이터 량이 많아지게 되어 아직도 많은 시간이 필요하게 된다.

이러한 문제를 해결하기 위해 3차원 동영상의 프레임간의 중복 데이터를 효과적으로 제거함으로써 3차원 비디오 홀로그래ム을 고속으로 생성할 수 있는 새로운 형태의 N-LUT 방식인 TR-N-LUT 기법이 제안되었다^[10].

그러나 기존 방법에서는 프레임간의 중복 데이터가 제거된 이후의 각 영상에서도 여전히 공간적인 중복성인 인터라인 데이터가 존재하기 때문에 각 프레임 영상 내에서의 공간적 중복 정보까지도 제거하게 되면 보다 효과적인 3차원 동영상 홀로그래ムの 합성이 가능하게 된다.

따라서, 본 논문에서는 3차원 동영상의 시·공간적 중복성을 동시에 고려한 새로운 형태의 3차원 비디오 홀로그래ム 생성기법인 TSR-N-LUT 기법을 제시하였다.

[Fig. 3]은 3차원 동영상의 시·공간적 중복성을 동시에 고려하여 3차원 비디오 홀로그래ム을 효과적으로 생성할 수 있는 새로운 TSR-N-LUT 기법의 동작과정을 나타낸 것으로 크게 네 부분으로 구성된다. 먼저, 3차원 동영상의 이전 프레임과 현재 프레임을 비교하여 두 프레임간의 차이를 추출한 다음, 두 프레임간의 차이가 50% 이상일 경우는

기존의 LR-N-LUT 기법을 이용하여 홀로그래ム을 생성하게 된다. 반면에 프레임간의 차이가 50% 미만일 경우 이전 프레임에서 사라진 부분과 현재 프레임에서 새로 생긴 부분에 대하여 각각 LR-N-LUT 기법을 이용하여 홀로그래ム을 생성한 후

이전 프레임의 홀로그래프에서 빼주고 더해주어 현재 프레임의 홀로그래프를 완성하게 된다. 여기서 각각 LR-N-LUT 기법을 이용하여 홀로그래프를 생성할 때에는 스캔 방향에 따라 이전 라인과 현재 라인의 변화량을 분석하여 역시 50%를 기준으로 LR-N-LUT 기법과 N-LUT 기법을 이용하여 홀로그래프를 생성하고 이를 마지막 라인까지 수행하게 된다. 최종적으로 생성된 홀로그래프를 전송하고 다음 프레임 계산을 위해 임시 버퍼에 보관하게 된다. 그리고 이러한 합성과정을 동영상 전체 프레임에 대하여 반복하게 된다.

3.1. 3차원 동영상의 시·공간적 중복정보 추출

[Fig. 4]는 3차원 동영상의 시·공간적인 중복 데이터를 추출하는 과정을 설명하기 위한 그림으로 [Fig. 4(a), (b)]는 이웃한 두 프레임 영상을 각각 나타낸다. [Fig. 4]에서는 편의상 같은 평면상의 영상으로 가정하고 밝기 정보만을 표현하였다. [Fig. 4(c)]는 [Fig. 4(a)]와 [Fig. 4(b)]를 비교하여 변화가 생긴 부분만을 표현한 것이다. 즉, 시간적 중복데이터가 제거된 영상이다. 이를 보면 전체 50픽셀 중 23픽셀이 이전 프레임과 다른 정보를 갖는 것을 볼 수 있다. 따라서 이전 프레임 홀로그래프에서 23픽셀만 보정을 해주면 홀로그래프를 생성할 수 있게 된다. 즉, 이전 정보를 빼고 현재 정보를 더해야 하기 때문에 46(= 23 × 2)번의 연산이 필요하게 된다.

[Fig. 4(d)]는 시간적 중복 데이터가 추출된 영상으로부터 인터라인 간의 공간적 중복데이터를 추출한 영상을 나타낸 것이다. 첫 번째 라인과 두 번째 라인을 보면 라인들 간에 같은 값을 갖는 픽셀이 없어 기존의 N-LUT 방식을 이용하여 홀로그래프를 생성한다.

하지만 세 번째 라인은 첫 번째, 여섯 번째, 일곱 번째 픽셀이 두 번째 라인과 다른 값을 갖고 나머지는 같은 값을 갖는 것을 볼 수 있다.따라서 첫

번째, 여섯 번째 픽셀은 두 번째 라인의 값을 빼주고 일곱 번째 픽셀 값은 더해지게 되어 3번의 연산으로 세 번째 라인의 홀로그래프를 생성할 수 있다. 다음, 네 번째 라인은 여덟 번째 픽셀 값만 변하기 때문에 50을 제거하고 100을 더하는 2번의 연산으로 홀로그래프를 생성할 수 있다. 그리고 마지막 라인은 많은 변화가 존재하기 때문에 기존의 N-LUT 방법으로 홀로그래프를 생성하여 7번의 연산을 수행하게 된다. 따라서 시·공간적 중복성이 적용된 [Fig. 4(d)]의 영상에 대한 홀로그래프를 생성하기 위해서는 20(=3(N-LUT) + 5(N-LUT) + 3(LR-N-LUT) + 2(LR-N-LUT) + 7(N-LUT))번의 연산을 수행하게 된다.

따라서 전체 50픽셀 중 20픽셀만 연산하면 되므로 30픽셀의 연산량이 감소된 것을 알 수 있다.

[Fig. 5]는 3차원 동영상에서 추출된 일부 프레임 영상을 나타낸 것이다. 이 3차원 동영상은 두 부분으로 나뉘어 있는데 첫 번째 부분(Part I; 1 ~ 30 프레임)은 뒤쪽에 있는 집 앞을 자동차가 오른쪽에서 왼쪽으로 이동하는 영상이고, 두 번째 부분(Part II; 31 ~ 60프레임)은 측면 상단에서 본 모습의 영상으로 각각 500 × 400의 해상도를 갖는다.

각각의 영상에서 볼 수 있듯이 집은 정지해 있고 자동차는 이동하는 것을 볼 수 있다. [Fig. 6]은 이전 프레임과 비교하여 차이가 있는 부분을 나타낸 영상으로 차가 오른쪽에서 왼쪽으로 이동하면서 변화된 부분을 추출하였다. [Fig. 6(a)]와 [Fig. 6(b)]를 보면 집은 고정되어 있고 자동차만 이동하기 때문에 변화하는 부분은 자동차의 일부만이 된다.

[Fig. 6(c)]의 경우는 영상의 씬(scene)이 변하는 부분으로 집주변을 이동하는 자동차의 정면에서 본 영상으로부터 우측 상단에서 본 영상으로 바뀌기 때문에 많은 부분이 바뀌게 된다. [Fig. 6(d)]의 경우를 보면 다시 자동차의 일부 영역만이 변화하는 것을 볼 수 있다.

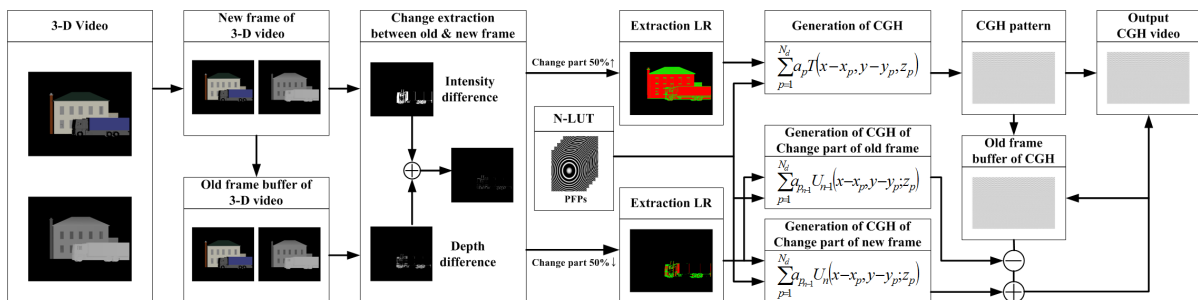


Fig. 3. Block diagram of the TSR-N-LUT method for generation of CGH patterns

[Fig. 7]은 [Fig. 6]의 시간적 중복 데이터가 추출된 영상을 이용하여 위에서 아래 방향으로 인터라인 간 중복 데이터를 추출한 결과를 나타낸 것이다.

인터라인 중복성 영상에서 녹색은 현재 라인의 현재 포인트가 이전 라인과 다른 것으로 보정을 해주어야 하는 부분이고 붉은색은 이전 라인과 같은 값을 갖기 때문에 계산을 하지 않아도 되는 부분을 나타낸 것이다.

[Fig. 7(a)]의 영상을 보면 첫 번째 프레임 영상이므로 모든 포인트에 대한 홀로그램 연산을 하여야 한다. 따라서 LR-N-LUT 기법을 이용하여 홀로그램을 생성하게 된다. [Fig. 7(c)]의 경우는 씬이 바뀌는 부분으로 많은 변화가 있게 되므로 역시 LR-N-LUT 기법을 이용하여 홀로그램을 생성한다.

250	250	250	150	150	50	250	250	250	150
250	150	50	50	250	150	250	150	50	50
50	250	250	250	150	150	50	250	250	250
50	50	250	250	150	150	50	50	250	250
50	250	250	250	150	150	50	250	250	250

(a)

250	150	250	100	100	50	250	250	250	150
150	150	50	50	250	100	250	50	150	150
50	250	250	250	150	150	150	50	150	150
50	50	250	250	150	150	150	100	150	150
50	150	50	50	150	150	200	50	50	50

(b)

	150		100	100					
150					100		50	150	150
						150	50	150	150
						150	100	150	150
	150	50	50			200	50	50	50

(c)

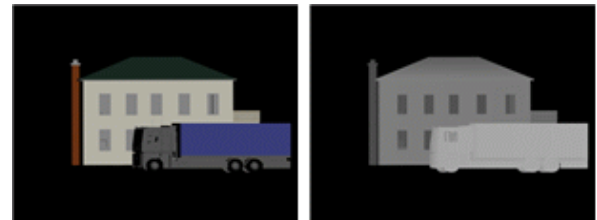
	150		100	100					
150					100		50	150	150
						150			
							100		
	150	50	50			200	50	50	50

(d)

(a) Previous frame (b) Current frame (c) Extracted temporal redundancy data (d) Extracted temporal redundancy data

Fig. 4. Extracted of TSR redundancy data of 3-D video images:

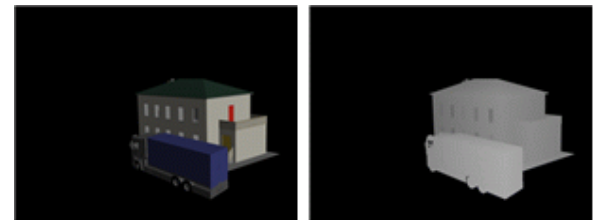
하지만 [Fig. 7(b)]와 [Fig. 7(d)]의 경우는 29~30 프레임으로 넘어가는 부분과 59~60 프레임으로 넘어가는 부분으로 변화가 많지 않은 것을 볼 수 있다. 따라서 이전 영상에서 사라진 부분과 현재 영상에서 새로 생긴 부분에 대해 LR-N-LUT 기법을 이용하여 홀로그램을 생성하여 이전 프레임의 홀로그램에 더하고 빼 줌으로써 현재 프레임의 홀로그램을 생성하게 된다.



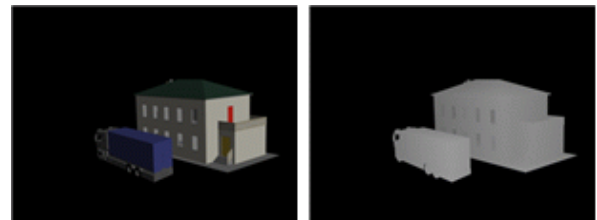
(a) 1st frame



(b) 30th frame



(c) 31st frame



(d) 60th frame

Fig. 5. 'Intensity image' and 'Depth image' of 3-D video frames(1st, 30th, 31st, 60th)



(a) 1st~2nd frame

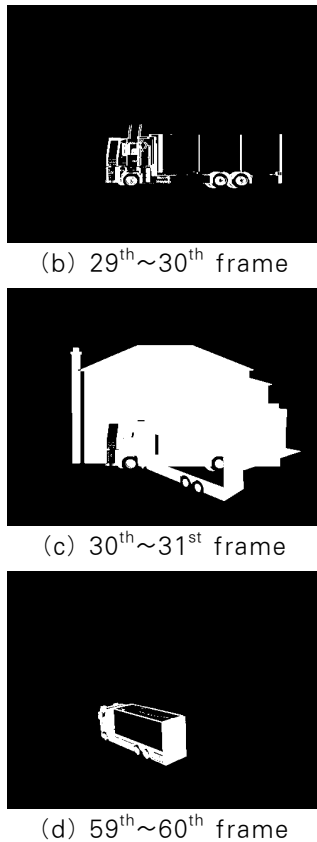


Fig. 6. Extracted temporal redundancy data of 3-D video images

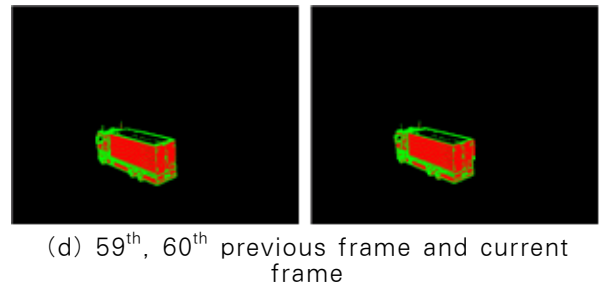
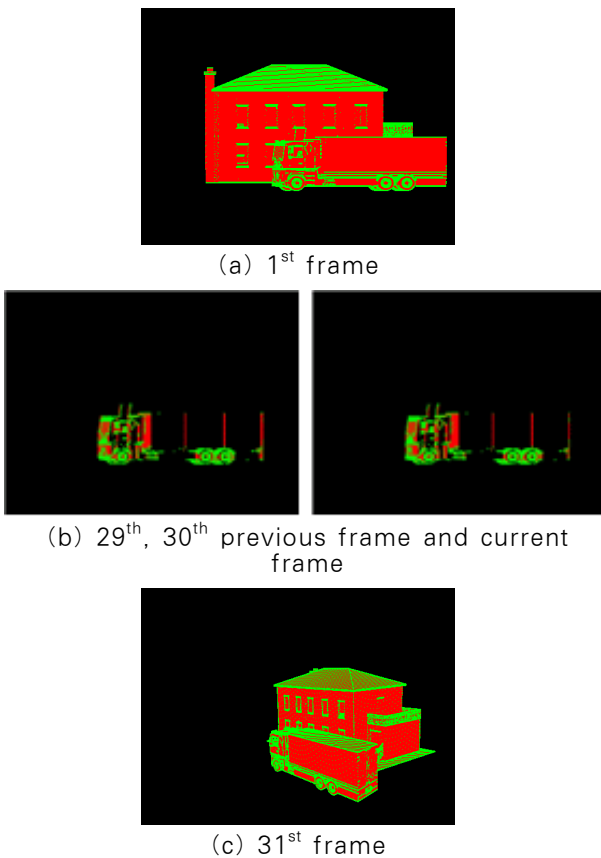


Fig. 7. Interline redundancy maps extracted with a vertical scanning method

3.2. TSR-N-LUT의 구성

앞에서 언급했지만 본 논문에서는 홀로그램을 생성하기 위하여 임의의 물체를 자체 발광하는 물체 점들의 집합으로 모델링하였다. 그리고 물체의 각 점들은 a_p 의 세기값을 가지고 수직, 수평, 깊이 방향에서 x_p, y_p, z_p 에 위치한다고 가정하였다. 이 때, n 번째 영상의 복소 진폭 $O_n(x, y)$ 는 다음과 같이 나타낼 수 있다.

$$O_n(x, y) \equiv \sum_{p=1}^{N_n} \frac{a_p}{r_p} \exp[j(kr_p + \phi_p)] \quad (10)$$

여기서 N_n 는 n 번째 프레임의 계산할 물체 점들의 수를 나타내고, 파수(wave number) 벡터 k 는 $k = 2\pi/\lambda$ 로 정의되며 λ 는 자유공간상에서의 빛의 파장을 나타낸다. 본 논문에서는 홀로그램을 생성하기 위하여 N-LUT 기법을 사용하였다. 이러한 N-LUT 기법에서는 각 평면의 중앙에 있는 점에 해당하는 대표요소프린지패턴 만을 미리 만들어 룩업 테이블에 저장을 하고, 이를 불러와 x, y 방향으로 각각 이동시켜 평면상의 모든 점들을 표현하게 된다. 즉, 각각의 대표 요소프린지패턴 $T(x, y; z_p)$ 는 임의의 깊이 z_p 에 대한 기준 세기의 프레넬 존 플레이트로 식 (8)과 같다.

제안하는 방법에서 첫 번째 프레임의 영상에 대해서는 기존의 LR-N-LUT 방식을 이용하여 홀로그램을 생성한다. 그리고 첫 번째 프레임 영상과 생성된 홀로그램 패턴은 다음 프레임의 연산을 위하여 각각의 임시 버퍼에 보관하게 된다. 다음 단계로서, 현재 프레임과 버퍼에 저장된 이전 프레임의 3차원 영상을 비교하여 두 영상간의 차이를 추출해 낸다. 추출한 영상의 변화율이 50% 이상이면 기존의 LR-N-LUT 방식을 이용하고 변화율이 50% 이하일 경우 이전 프레임에서 사라진 부분에 대한 영상과 현재 프레임에서 추가된 부분에 대한 영상에서 각각 인터라인 중복성 정보를 추출한다. 그리고 이

전 프레임의 사라진 부분과 현재 추가된 부분에 대하여 LR-N-LUT 기법을 이용하여 홀로그램을 생성하고 이전 프레임의 홀로그램에 각각 빼고 더해서 현재 프레임의 홀로그램을 생성하고 이는 다음과 같이 표현된다.

$$I_n = I_{n-1} - I_{n-1_{disp}} + I_{n_{add}} \quad (11)$$

여기서 I_{n-1} 은 이전 프레임의 홀로그램을, $I_{n-1_{disp}}$ 는 이전 프레임에서 사라진 부분에 대한 홀로그램을, $I_{n_{add}}$ 는 현재 프레임에서 새로 생긴 부분에 대한 홀로그램을 각각 나타낸다. 그리고 각 홀로그램을 생성할 때에는 LR-N-LUT 기법을 사용하게 되고 이는 다음과 같이 표현할 수 있다.

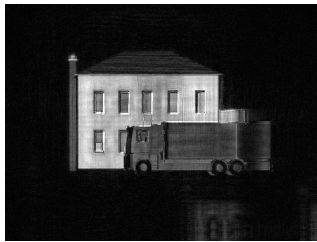
$$I_n(x, y) = \sum_{l=1}^{N_l} \left[I_{l-1}(x, y) - \sum_{p=1}^{N_d} a_{p_{l-1}} U_{l-1}(x - x_p, y - y_p; z_p) \right. \\ \left. + \sum_{p=1}^{N_d} a_{p_l} U_l(x - x_p, y - y_p; z_p) \right] \quad (12)$$

여기서, N_l 은 입력 영상의 전체 라인의 수를, I_{l-1} 은 이전 라인의 홀로그램 영상을, N_d 는 n 번째 라인과 $n-1$ 번째 라인 간 영상의 차이에 해당하는 포인트의 수를 나타낸다. 또한, $U_l(x, y; z_p)$ 는 1번째 라인의 PFP를 나타내고 다음과 같이 표현된다.

$$U_l(x, y; z_p) = \begin{cases} T(x, y; z_p) & \text{for changed part} \\ 0 & \text{for unchanged part} \end{cases} \quad (13)$$

마지막으로, 생성된 홀로그램 패턴은 비디오 홀로그램으로 출력이 되며 동시에 임시 버퍼에 저장되고 이러한 프로세스를 전체 동영상에 대하여 반복하여 제안된 방식의 3차원 동영상 비디오 홀로그램을 생성하게 된다.

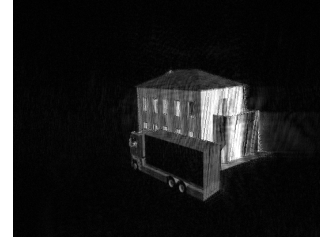
3.3. TSR-N-LUT 기반의 홀로그램 생성 및 복원



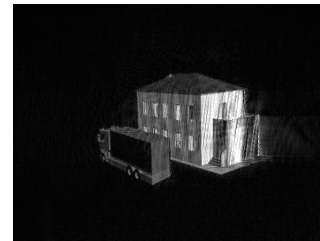
(a) 1st frame



(b) 30th frame



(c) 31st frame



(d) 60th frame

Fig. 8. Reconstructed of conventional TR-N-LUT method

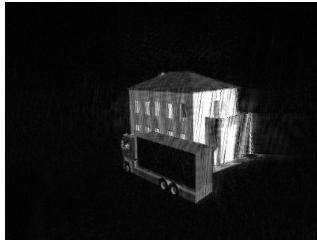
본 논문에서는 제안하는 방법을 검증하기 위하여 $10\mu\text{m}$ 의 픽셀 크기로 $1,200 \times 1,200$ 크기의 홀로그램을 생성하였다. [Fig. 8]은 기존 TR-N-LUT 기에 의해 생성된 홀로그램을 복원한 영상이고 [Fig. 9]는 제안된 기법을 이용하여 홀로그램을 생성하여 복원한 영상이다. 이 두 영상을 비교하여 보면 복원된 영상에는 큰 차이가 없는 것을 볼 수 있다.



(a) 1st frame



(b) 30th frame



(c) 31st frame



(d) 60th frame

Fig. 9. Reconstructed of proposed method

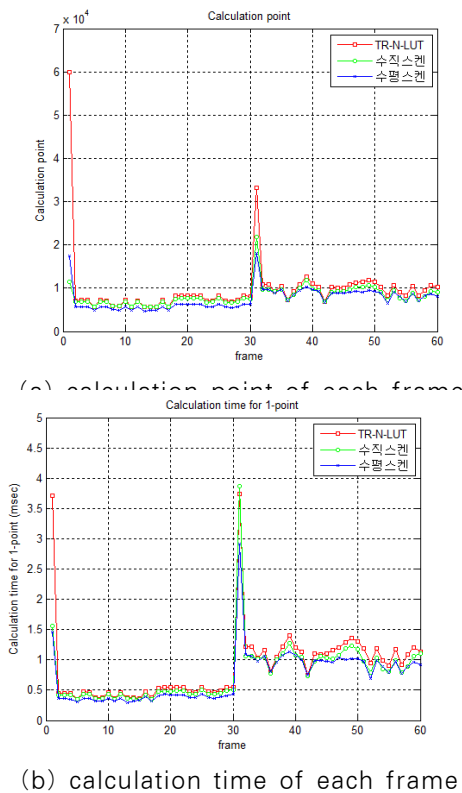


Fig. 10. Comparison with conventional and proposed method

[Fig. 10]은 각각의 방법에 대한 각 프레임별 계산 포인트 수와 프레임별 포인트 당 계산시간을 나타낸 것이다. 이를 보면 기존의 TR-N-LUT 기법에 비해 제안된 방법이 계산할 물체 포인트 수도 줄어들고 포인트 당 계산시간 또한 줄어드는 것을 볼 수 있다. 또한, 첫 번째 프레임의 경우는 다른 방식과 같이 모든 물체 포인트의 연산을 수행하므로 많

은 시간이 걸리고 씬이 바뀌는 31번째 프레임 역시 모든 포인트를 연산하기 때문에 많은 시간이 걸리게 된다. 하지만 그 이외의 프레임에서는 변화량이 많지 않기 때문에 훨씬 적은 시간이 걸리는 것을 볼 수 있다.

Table 1. Comparison results between the conventional and proposed methods

		TR-N-LUT	TSR-N-LUT (Vertical scan)	TSR-N-LUT (Horizontal scan)
1st - 30th frame	Number of calculated points	8,831.97 (100%)	6,890.97 (78.02%)	6,036.73 (68.35%)
	Computation time for 1-point (ms)	0.57 (100%)	0.46 (81.40%)	0.41 (71.07%)
31st - 60th frame	Number of calculated points	10,873.30 (100%)	9,535.67 (87.70%)	8,994.03 (82.72%)
	Computation time for 1-point (ms)	1.21 (100%)	1.10 (91.20%)	1.03 (84.87%)
Total	Number of calculated points	9,852.63 (100%)	8,213.32 (83.36%)	7,515.38 (76.28%)
	Computation time for 1-point (ms)	0.89 (100%)	0.78 (88.06%)	0.72 (80.45%)

[Table 1]은 기존 TR-N-LUT 기법과 제안된 TSR-N-LUT 기법에 대해 평균 계산 포인트 수와 홀로그램 연산 시간을 비교한 결과를 나타낸 것이다. 본 실험은 Intel i5 Core Quad 2.5 GHz, 4GB 메모리와 Microsoft Windows7 기반의 Matlab 2009a를 사용하여 수행하였다 수직스캔의 경우 1st ~ 30th 프레임의 연산 포인트는 TR-N-LUT 기법의 8,831.97 포인트에서 6,980.97 포인트로 제안된 방식이 기존의 방법보다 21.98%의 연산 포인트 수가 줄어드는 것을 볼 수 있다. 또한, 31st ~ 60th 프레임에서는 10,873.30 포인트에서 9,535.67 포인트로 줄어들고 전체 3차원 동영상에 대해서는 평균적으로 9,852.63 포인트에서 8,213.32 포인트로 감소되어 약 16.64%의 연산 포인트가 줄어드는 것을 볼 수 있다. 포인트당 홀로그램 생성시간은 기존 방법의 0.89 ms에서 0.78 ms로 11.94%가 줄어드는 것을 볼 수 있다. 마찬가지로 수평스캔의 경우도 23.72%의 연산 포인트가 줄어들고 포인트당 연산시간은 19.55%가 각각 줄어드는 것을 볼 수 있다. 따라서 기존의 방법에 비해 새로이 제안된 TSR-N-LUT 방식은 홀로그램 계산을 위한 물체 포인트 수의 감소 뿐 만 아니라 각 포인트의 연산시간도 줄일 수 있어 실제 응용에서 효과적으로 사용이 가능하다고 분석된다.

IV. 결 론

본 논문에서는 3차원 동영상의 시공간적 내의 인터라인 정보 중복성을 이용한 새로운 홀로그램 생성 방법인 TSR-N-LUT을 제시하고 실험을 통해 제안된 기법의 성능을 분석하였다. 제안된 기법에서는 3차원 동영상에 대한 시간적 중복 데이터를 제거한 다음, 남은 영상에 대해서도 라인간의 공간적 중복 데이터를 또한 제거함으로써 홀로그램 계산을 위한 물체 포인트 수가 크게 줄어들게 되고 결과적으로 전체적인 홀로그램 생성속도가 향상되었다. 실험결과 제안된 TSR-N-LUT 기법은 기존 방법에 비해 평균적으로 홀로그램 계산을 위한 물체 포인트 수가 23.72% 감소하고, 동시에 홀로그램 생성속도도 19.55% 개선됨을 보임으로써 제안된 기법의 실제 응용 가능성을 제시하였다.

참 고 문 헌

[1] E.-S. Kim, S.-H. Lee(Translated by), Basic of 3-D video(3차원 영상의 기초), Kidari, 1998.

[2] A. W. Lohmann and D. P. Paris, "Binary fraunhofer holograms generated by computer," *Appl. Opt.*, vol. 6, pp. 1739-1748, 1967.

[3] M. Paturzo, P. Memmolo, A. Finizio, R. Näsänen, T. J. Naughton and P. Ferraro, "Holographic display of synthetic 3D dynamic scene", *3D Research*, 1, pp. 31-35, Jun.2010.

[4] P. W. M. Tsang, J. P. Liu, K. W. K. Cheung and T. -C. Poon, "Modern methods for fast generation of digital holograms", *3D research*, 1, pp. 11-18, May.2010.

[5] C. J. Kuo and M. H. Tsai, *Three-Dimensional Holographic Imaging*, John Wiley & Sons, 2002.

[6] M. Lucente, "Interactive computation of holograms using a look-up table," *J. Electron. Imag.*, vol. 2, pp. 28-34, 1993.

[7] S.-C. Kim, E.-S. Kim, "Fast Generation and Reconstruction of Digital Holograms Using a Novel Look-up Table", *The Korean Institute of Communications and Information Sciences (KICS)*, vol. 33, no. 3, pp. 255-261, Mar.2008

[8] S.-C. Kim, K.-D Na and E.-S. Kim,

"Accelerated computation of computer-generated holograms of a 3-D object with N×N-point principle fringe patterns in the novel look-up table method", *Opt. Lasers Eng.*, (계제예정).

[9] S.-C. Kim and E.-S. Kim, "Fast computation of hologram patterns of a 3-D object using run-length encoding and novel look-up table methods", *Appl. Opt.*, vol. 48, no. 6, pp. 1030-1041, Feb.2009.

[10] S.-C. Kim, E.-S. Kim, "Fast Generation of 3-D Video Holograms using a Look-up Table and Temporal Redundancy of 3-D Video Image", *The Korean Institute of Communications and Information Sciences (KICS)*, vol. 34, no. 10, pp. 1076-1085, Oct.2009.

[11] S.-C. Kim, W.-Y. Choe and E.-S. Kim, "Accelerated computation of hologram patterns by use of interline redundancy of 3-D object images", *Opt. Eng.*, vol. 50, no. 9, pp. 091305, Apr.2011.

[12] S.-C. Kim, J.-M. Kim and E.-S. Kim, "Effective memory reduction of the novel look-up table with one-dimensional sub-principle fringe patterns in computer-generated holograms.", *Opt. Express*, vol. 20, no. 11, pp. 12021-12034, May.2012.

김 동 욱 (Dong-wook Kim)



2011년 2월 광운대학교 전자공학과 졸업
 2011년 3월~현재 광운대학교 전자공학과 석사과정
 <관심분야> 3차원 디스플레이, 홀로그래픽 디스플레이, 광공학

구 정 식 (Jung-Sik Koo)

한국통신학회지 Vol.29, No. 4C 참조

김 승 철 (Seung-Cheol Kim)

한국통신학회지 Vol.34, No. 10 참조

김 은 수 (Eun-Soo Kim)

한국통신학회지 Vol.24, No. 9A 참조