

룩업테이블과 3차원 동영상의 시간적 중복성을 이용한 3차원 비디오 홀로그램의 고속 생성

정회원 김 승 철*, 종신회원 김 은 수*

Fast Generation of 3-D Video Holograms using a Look-up Table and Temporal Redundancy of 3-D Video Image

Seung-Cheol Kim* *Regular Member*, Eun-Soo Kim* *Lifelong Member*

요 약

본 논문에서는 3차원 동영상의 시간적 중복성과 룩업 테이블을 이용한 효과적인 3차원 동영상 홀로그램 생성 방법을 제안한다. 기존의 N-LT를 이용한 홀로그램 생성 기법은 미리 생성된 요소 프린지 패턴을 단순 이동 시켜 물체에 대한 홀로그램을 고속으로 생성할 수 있게 하였다. 하지만, 3차원 동영상 홀로그램의 경우는 정지영상에 대한 홀로그램에 비해 방대한 양으로 인하여 실시간 홀로그램을 생성하는데 많은 시간이 필요하게 되어 실질적인 제한이 있었다. 반면 이러한 3차원 동영상은 인접한 프레임은 유사한 정보를 갖는다는 시간적 중복성을 갖고 있고, 이러한 특성을 동영상 압축에 이용하고 있다. 따라서 본 논문에서는 이러한 3차원 동영상의 시간적 중복성을 이용하여 계산하여야 하는 데이터의 양을 현저하게 줄임으로써 3차원 동영상에 대한 홀로그램을 고속으로 생성할 수 있는 새로운 방법을 제안하고 실험을 통하여 검증하였다.

Key Words : 3D Display, Holography, CGH(Computer Generated Holography), Look-up Table

ABSTRACT

In this paper, a new method for efficient computation of CGH patterns for 3-D video images is proposed by combined use of temporal redundancy and look-up table techniques. In the conventional N-LT method, fringe patterns for other object points on that image plane can be obtained by simply shifting these pre-calculated PFP (Principle Fringe Patterns). But there have been many practical limitations in real-time generation of 3-D video holograms because the computation time required for the generation of 3-D video holograms must be massively increased compared to that of the static holograms. On the other hand, as ordinary 3-D moving pictures have numerous similarities between video frames, called by temporal redundancy, and this redundancy is used to compress the video image. Therefore, in this paper, we proposed the efficient hologram generation method using the temporal redundancy of 3-D video image and N-LT method. To confirm the feasibility of the proposed method, some experiments with test 3-D videos are carried out, and the results are comparatively discussed with the conventional methods in terms of the number of object points and computation time.

1. 서 론

3차원 입체영상 기술은 향후 기존의 2차원 디지

털 방송, 통신을 대신할 수 있는 차세대 실감 영상 미디어로 기대되고 있다. 또한, 원격의료, 가상현실, 영화, 게임 산업 등에서도 큰 영역을 크게 차지할

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2009-C1090-0902-0018).

* 광운대학교 전자공학과 차세대 3D 디스플레이 연구센터({sckim, eskim}@kw.ac.kr)

논문번호 : KICS2009-08-332, 접수일자 : 2009년 8월 5일, 최종논문접수일자 : 2009년 9월 16일

것으로 예상되고 있다.

그동안 다양한 방법의 3차원 영상 디스플레이 방식이 제안되거나 실제로 여러 응용에서 활용되고 있다¹¹⁾. 특히, 홀로그래피 기술은 원리적으로 완전 3차원 입체영상 기술이기 때문에 개발 초부터 3차원 디스플레이 기술로 활용하기 위한 많은 연구가 이루어졌으나 여러 가지 현실적인 문제로 그 응용이 실질적으로 제한되어 왔다¹²⁾. 즉, 기존의 광 홀로그램(optical hologram)은 가 간섭성 광인 레이저 광을 사용하여 만들어지기 때문에 실험공간이 암실로 제한되고, 진폭과 위상정보를 저장하기 위한 과정에서는 작은 움직임에도 간섭무늬가 파괴될 수 있기 때문에 매우 안정적인 광학 시스템이 요구된다. 이러한 문제를 해결하는 새로운 접근방법으로 컴퓨터를 이용하여 홀로그램 패턴을 생성하는 컴퓨터 형성 홀로그램(CGH: computer-generated hologram) 기법이 제안되었다¹³⁾.

일반적으로 임의 물체에 대한 디지털 홀로그램 패턴을 계산할 때 빛의 회절을 계산하는 광선추적(ray-tracing) 방식이 주로 사용되어 왔다. 이 방식에서는 물체를 점들의 집합으로 보고 각각의 물체 점들에 대한 홀로그램 패턴을 모두 계산하여 합산하게 된다. 따라서 이 방법은 과도한 계산량이 요구되기 때문에 복잡한 물체에 대한 실시간적 홀로그램 합성을 어렵게 한다.

이러한 문제점을 극복하기 위하여 룩업테이블(LT: look-up table)을 이용한 새로운 디지털 홀로그램 계산 방법이 제안되었다¹⁴⁾. 이 방법은 모든 가능한 물체 포인트에 대한 요소 프린지 패턴(EFP: elemental fringe pattern)을 미리 계산하여 데이터베이스(data base)로 저장하게 된다. 따라서 임의 물체에 대한 홀로그램 계산은 그 물체를 구성하고 있는 각 포인트에 대한 요소 프린지 패턴들을 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 크게 증가하게 되는 단점을 가지고 있다.

최근 기존의 LT 방식의 장점인 연산 속도는 유지하면서 기존 LT 방식의 단점인 방대한 메모리 공간을 줄일 수 있는 N-LT 방식이 제안되었다¹⁵⁾. 이 방법에서는 3차원 물체를 각각의 깊이에 해당하는 평면들의 집합으로 간주를 하고, 이 평면은 각각

자체 발광하는 점들의 집합으로 간주를 한다. 그리고 각 평면의 중앙에 있는 점에 해당하는 요소 프린지 패턴(PFP; Principle Fringe Pattern)을 미리 만들어 룩업테이블에 저장을 하고, 이를 불러와 수평 방향으로 각각 이동시켜 평면상의 모든 점들을 표현하게 된다. 따라서 N-LT 방식에서는 물체의 깊이 방향에 대한 요소 프린지 패턴만을 계산하여 저장하게 되고, 물체의 한 깊이 방향이 결정되면 그 면에 존재하는 물체 포인트들의 요소 프린지 패턴들은 사전에 저장된 그 깊이의 요소 프린지 패턴을 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 LT 방식의 고속 생성 특성은 유지 하면서 메모리 용량을 획기적으로 줄일 수 있게 되었다.

한편, N-LT 방식을 이용하여 홀로그램을 생성할 때 포인트 수가 늘어나게 되면 이에 비례하여 계산 시간도 늘어나게 된다. 이러한 문제를 해결하기 위하여 3차원 입력 영상 내부의 인접한 화소들 간의 공간적 중복성을 이용하여 생성할 포인트 수를 줄여 컴퓨터 형성 홀로그램을 생성하는 방법이 제안되었다¹⁶⁾.

하지만 3차원 동영상에 대하여 홀로그램을 생성할 경우 각 프레임에 대해서는 데이터양이 크지 않지만 동영상의 프레임 수가 늘어남에 따라 계산해야 할 데이터가 방대해지게 된다. 따라서 3차원 동영상에 대한 실시간 홀로그램 생성에는 한계가 있게 된다.

반면, 일반적으로 2차원 동영상은 연속적인 2차원 정지영상들의 집합으로 볼 수 있다. 여기서 인접한 영상간의 영상들은 서로 비슷한 모양을 갖는다. 즉, 2차원 동영상은 시간적 중복성을 갖게 된다. 일반적으로 이러한 유사성은 DPCM(Differential Pulse Code Modulation) 기법을 이용하여 동영상의 압축에 적용된다. 즉, 첫 번째 영상에 대해서는 영상의 전 부분을 전송하고, 두 번째 영상부터는 이전영상과의 차이만을 전송하여 준다. 그리고 수신측에서는 첫 번째 영상으로부터 다음 영상과의 차이를 이용하여 현재의 영상을 복원하게 된다. 이는 3차원 동영상이라고 해서 달라지지 않는다. 단지 밝기 영상과 깊이 영상으로 확대될 뿐이다. 따라서 이러한 3차원 동영상에 대해 시간적 중복성을 이용하여 홀로그램을 생성하게 되면 더욱 효율적인 홀로그램

생성이 가능할 것이다.

따라서 본 논문에서는 3차원 동영상의 대하여 시간적 중복성 특성과 홀로그래프를 고속으로 생성할 수 있는 N-LT 기법을 이용하여 3차원 동영상에 대한 홀로그래프의 고속 생성 기법을 제안한다.

즉, 3차원 동영상의 이전 프레임과 현재의 프레임을 비교하여 두 프레임간의 차이를 추출하고 프레임간의 차이가 50% 이상일 경우는 기존의 N-LT 방법을 사용하여 홀로그래프를 생성한다. 하지만 차이가 50% 이하일 경우는 차이에 대한 이전 프레임과 현재 프레임의 홀로그래프를 생성하고 이전 프레임의 전체 영상에 대한 홀로그래프 패턴에서 각각 빼고 더하여 현재 프레임의 전체 영상에 대한 홀로그래프를 생성한다. 그리고 마지막으로 생성된 홀로그래프 패턴을 전송하고 다음 프레임 계산을 위해 임시 버퍼에 보관하게 된다. 따라서 계산해야 하는 데이터양이 현저히 줄어들게 되므로 홀로그래프를 생성하는 속도는 현저히 증가하게 되어 실시간 응용이 가능하게 된다.

따라서 본 논문에서 새로이 제안하는 시간적 중복성과 N-LT 기법을 이용한 디지털 홀로그래프 합성 방법의 실제 응용 가능성을 검증하기 위하여 집 앞을 지나가는 자동차 동영상을 사용하여 실험하고 그 결과를 분석하였다. 즉, 기존의 N-LT 방식과 본 논문에서 새로이 제안된 방식을 이용하여 동영상 홀로그래프를 각각 합성하고, 각 방식에서 요구되는 홀로그래프 생성시간을 비교 분석하였다.

II. 기존의 광선추적 및 록업테이블 방식

2.1 기존의 광선추적 방식

디지털 홀로그래프 패턴은 픽셀들의 좌표 값과 세기 값에 의해 컴퓨터 계산으로 생성된다. 즉, 디지털 홀로그래프 패턴을 생성하는 기하학적 구조는 그림 1과 같으며^[7], 그림 1에서 홀로그래프는 $z = 0$ 인

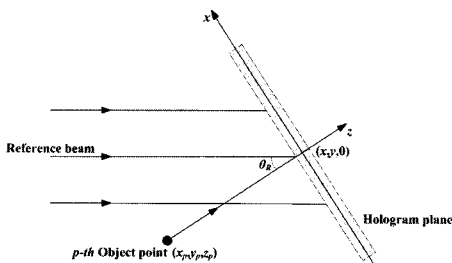


그림 1. 프레임 홀로그래프의 기하학적 구조

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.

이 때, 홀로그래프에서 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 나타낸다. 편위상 $\exp(jax)$ 는 식 (1)에서 포함시키지 않았다. 홀로그래프에서 p 번째 물체 점과 $(x, y, 0)$ 사이의 거리 r 은 식 (2)와 같이 주어진다.

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

또한, 평면파인 기준파의 복소 진폭 $R(x, y)$ 는 식 (3)과 같이 주어진다.

$$R(x, y) = a_R \exp[j(-kx \sin \theta_R)] \quad (3)$$

여기서 a_R 과 θ_R 는 기준파의 세기와 입사각을 각각 나타낸다. 홀로그래프 면에서의 전체적인 격자 세기는 식 (4)과 같이 물체파 $O(x, y)$ 와 기준파 $R(x, y)$ 사이의 간섭패턴으로 주어진다.

$$\begin{aligned} I(x, y) &= |O(x, y) + R(x, y)|^2 \\ &= |R(x, y)|^2 + |O(x, y)|^2 \\ &\quad + 2|R(x, y)||O(x, y)| \cos [kr_p + kx \sin \theta_R + \phi_p] \end{aligned} \quad (4)$$

식 (4)에서 첫 번째와 두 번째 항은 물체파와 기준파의 세기를 각각 나타내며, 세 번째 항은 홀로그래프 정보를 부분적으로 포함하고 있는 물체파와 기준파 사이의 간섭 패턴을 의미한다. 식 (5)에서 홀로그래프 정보는 오직 세 번째 항에만 포함되어 있기 때문에 홀로그래프 패턴 $I(x, y)$ 는 최종적으로 아래와 같이 표현될 수 있다.

$$I(x, y) = 2 \sum_{p=1}^N \frac{a_p}{r_p} \cos (kr_p + kx \sin \theta_R + \phi_p) \quad (5)$$

2.2 기존의 LT 방식

디지털 홀로그래프의 연산속도를 향상시키기 위한 새로운 접근방법으로 록업테이블(LT) 방식이 M.

Lucent에 의해 제안되었다⁴¹. 이 방식은 가능한 물체영상 공간 내의 모든 점 (x_p, y_p, z_p) 에 대하여 기준 세기의 요소 프린지 패턴 $I(x, y)$ 를 미리 계산하여 테이블로 만들어 놓은 후에, 물체에 따라 필요한 포인트에 대한 요소 프린지 패턴을 불러내어 사용하는 방식이다.

일반적으로 실세계의 공간은 연속적이다. 따라서 실세계의 공간은 무한한 점을 갖게 되고 록업테이블을 이용해 홀로그래프를 만들기 위해서는 무한개의 요소 프린지 패턴이 필요하게 된다. 하지만 인간의 시각 시스템의 능력은 제한되어 있기 때문에 물체 점들을 이산화 하여 홀로그래프를 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.⁴¹

LT 방식의 요소 프린지 패턴은 식 (5)로부터 생성해 낼 수 있다. 즉, 기준 세기를 갖는 요소 프린지 패턴 $T(x, y; x_p, y_p, z_p)$ 는 다음과 같이 표현된다.

$$T(x, y, x_p, y_p, z_p) \equiv \frac{1}{r_p} \cos [kr_p + kx \sin \theta_R + \phi_p] \quad (6)$$

여기서, r_p 는 p 번째 점과 $(x, y, 0)$ 사이의 거리로 식 (2)와 같이 주어진다.

이 방식에서는 홀로그래프를 계산할 때 식 (5)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 록업테이블을 이용하여 계산을 하게 된다. 따라서 LT 방식에서 홀로그래프 정보 $I(x, y)$ 는 최종적으로 식 (7)과 같이 주어지며, 식에서 N 은 물체 포인트 수를 나타낸다.

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

홀로그래프에서 위상(ϕ)은 이미지의 포인트 간의 상호작용을 나타내는 중요한 요소이다. 그러나 위상 성분은 사람의 눈으로는 볼 수 없고 또한 두 점간의 간격이 충분히 넓어 점의 크기보다 충분히 커질

때는 크게 영향을 주지 않는다. 따라서, 본 논문에서는 두 점간의 간격을 150 micron으로 하였기 때문에 위상의 영향을 크게 받지 않으므로 각 포인트의 위상은 0으로 가정하였다⁴¹.

2.3 기존의 N-LT 방식

최근 기존의 LT 방식의 장점인 연산 속도는 유지를 하면서 기존 LT 방식의 단점인 방대한 메모리 공간을 줄일 수 있는 N-LT 방식이 제안되었다. 이 방법에서는 3차원 물체를 각각의 깊이에 해당하는 평면들의 집합으로 간주를 하고, 이 평면은 각각 자체 발광하는 점들의 집합으로 간주를 한다. 그리고 각 평면의 중앙에 있는 점에 해당하는 PFP를 미리 만들어 록업테이블에 저장을 하고, 이를 불러와 x, y 방향으로 각각 이동시켜 평면상의 모든 점들을 표현하게 된다. 즉, 각각의 요소 프린지 패턴 $T(x, y; z_p)$ 은 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 $(x, y, 0)$ 사이의 거리로 식 (2)와 같이 주어진다. 따라서 N-LT 방식에서는 물체의 깊이 방향에 대한 요소 프린지 패턴만을 계산하여 저장하게 되고, 물체의 한 깊이 방향이 결정되면 그 면에 존재하는 물체 포인트들의 요소 프린지 패턴들은 사전에 저장된 그 깊이의 요소 프린지 패턴을 각 물체 포인트까지 이동시켜 프린지 패턴을 계산하여 그 깊이 평면에서의 홀로그래프 패턴을 계산하게 된다. 같은 방법으로 모든 물체 깊이 평면에서 각각의 홀로그래프를 계산하여 합산함으로써 물체 전체에 대한 홀로그래프 패턴을 계산하게 된다. 따라서 N-LT 방식에서 홀로그래프 정보 $I(x, y)$ 는 다음과 같이 표현할 수 있다.

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

III. 동영상의 시간적 중복성

3.1 2차원 동영상의 시간적 중복성

일반적으로 2차원 동영상은 연속적인 2차원 정지 영상들의 집합으로 볼 수 있다. 여기서 인접한 영상간의 영상들은 서로 비슷한 모양을 갖는다. 즉, 영

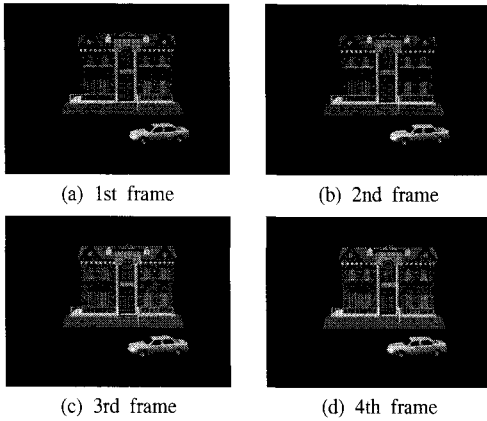


그림 2. 2차원 동영상에서 인접한 네 영상

상의 일부 부분은 그대로 멈추어 있고 일부 부분은 조금씩 움직이게 되어 두 영상 간에는 상당한 유사성을 갖게 되고 이러한 유사성을 시간적 중복성이라 한다.

그림 2는 2차원 동영상에서 1프레임 간격으로 추출된 네 장의 정지영상을 나타낸다. 이미 언급한 바와 같이 네 장의 영상이 거의 동일한 것처럼 보이지만 집의 경우는 정지되어 있고, 자동차가 오른쪽에서 왼쪽으로 이동하고 있는 것을 볼 수 있다. 일반적으로 이러한 유사성은 DPCM 기법을 이용하여 동영상의 압축에 적용된다. 즉, 첫 번째 영상에 대해서는 영상의 전 부분을 전송하고, 두 번째 영상부터는 이전영상과의 차이만을 전송하여 준다. 그리고 수신측에서는 첫 번째 영상으로부터 다음 영상과의 차이를 이용하여 현재의 영상을 복원하게 된다.

3.2 3차원 동영상의 시간적 중복성

2차원 동영상과 유사하게 3차원 동영상은 연속적인 밝기 영상과 깊이 영상을 갖는 3차원 정지영상들의 집합으로 볼 수 있다. 2차원 영상과 같이 인접한 3차원 영상간의 영상들은 서로 비슷한 모양을 갖는다. 즉, 영상의 일부 부분은 그대로 멈추어 있고 일부 부분은 조금씩 움직이게 되어 두 영상 간에는 상당한 유사성을 갖게 되어 2차원 영상과 같이 3차원 영상도 역시 시간적 중복성을 갖는다.

그림 3은 3차원 동영상에서 1프레임 간격으로 추출된 네 장의 밝기 영상과 깊이 영상을 나타낸다. 이미 언급한 바와 같이 네 쌍의 영상이 거의 동일한 것처럼 보이지만 뒤쪽에 있는 집의 경우는 정지되어 있고, 앞쪽에 있는 자동차가 오른쪽에서 왼쪽으로 이동하고 있는 것을 볼 수 있다. 즉, 3차원 동영상 역시 시간적 중복성을 갖는 것을 볼 수 있다.

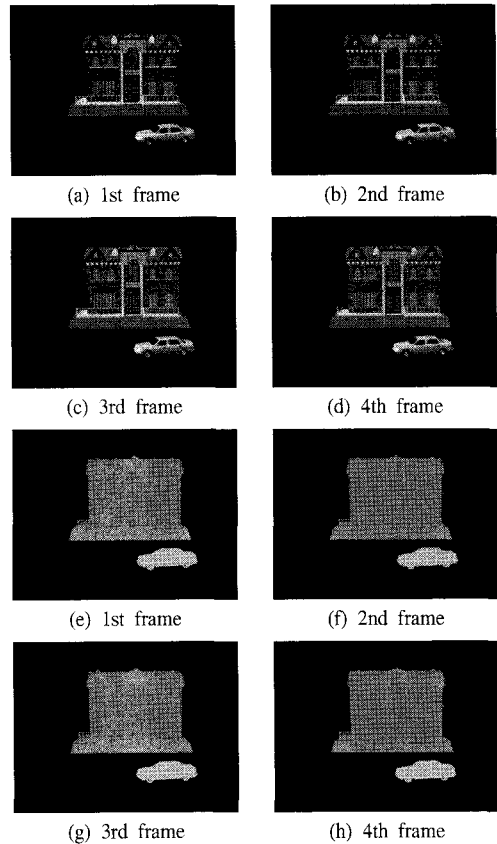


그림 3. 3차원 동영상에서 인접한 네 영상

따라서 이러한 3차원 동영상에 대하여 시간적 중복성을 적용하게 되면 3차원 동영상 홀로그래프를 생성하는데 있어 더욱 효과적이 될 것이다.

IV. 새로이 제안된 컴퓨터 형성 홀로그램 생성 방식

기존의 LT 방식에서는 가능한 물체영상의 모든 포인트에 대해 미리 계산 해놓은 요소 프린지 패턴을 이용함으로써 홀로그램 합성시 엄청난 속도의 증가를 가져왔다. 그러나 이 방식의 가장 큰 단점은 미리 계산해놓은 요소 프린지 패턴의 양이 워낙 많고 따라서 이를 저장할 LT의 메모리 또한 엄청나게 증가하게 된다. 예를 들어, LT 방식에서 물체공간이 100(가로) × 100(세로) × 100(깊이) 이고, 각 요소 프린지 패턴의 용량이 1MB라고 가정하면 전체 록업테이블의 메모리 용량은 $1MB \times 100 \times 100 \times 100 = 1TB$ 에 이르게 된다.

이러한 문제점을 해결하기 위해 제안된 N-LT 방식의 경우 계산 속도를 유지하며 필요로 하는 메모리 공간은 현저히 줄어들게 된다. 하지만 계산 속도

가 현저히 줄어든다 하더라도 동영상의 경우는 초당 30프레임의 영상이 존재하기 때문에 계산해야 할 양이 많아지게 되어 많은 시간이 필요하게 된다. 따라서 본 논문에서는 기존의 N-LT 방식과 3차원 동영상의 시간적 중복성을 적용하여 3차원 동영상에 대한 홀로그래మ్을 고속으로 생성할 수 있는 방법을 제안하였다.

그림 4는 새로이 제안된 3차원 동영상의 시간적 중복성과 N-LT 기법을 이용하여 3차원 동영상에 대한 홀로그래మ్을 효과적으로 생성할 수 있는 방법의 전체 블록 다이어그램을 나타낸 것으로 크게 세 부분으로 나뉜다. 첫째로, 3차원 동영상의 이전 프레임과 현재의 프레임을 비교하여 두 프레임간의 차이를 추출한다. 두 번째로, 프레임간의 차이가 50% 이상일 경우는 기존의 N-LT 방법을 사용하여 홀로그래మ్을 생성한다. 마지막으로 생성된 홀로그래మ్을 전송하고 다음 프레임 계산을 위해 임시 버퍼에 보관한다. 반면 차이가 50% 이하일 경우는 차이에 대한 이전 프레임과 현재 프레임의 홀로그래మ్을 생성하고 이전 프레임의 전체 영상에 대한 홀로그래మ్ 패턴에서 각각 빼고 더하여 현재 프레임의 전체 영상에 대한 홀로그래మ్을 생성한다. 마지막으로 역시 생성된 홀로그래మ్ 패턴을 전송하고 다음 프레임 계산을 위해 임시 버퍼에 보관한다. 그리고 이러한 절차를 동영상의 전체 프레임에 대하여 반복한다.

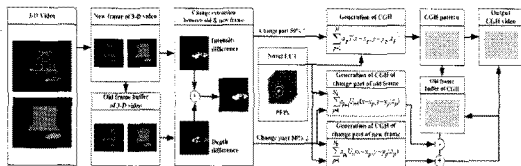


그림 4. 제안된 N-LT 기반의 3차원 동영상 홀로그래మ్ 생성 및 복원과정의 전체 블록 다이어그램

4.1 3차원 비디오 홀로그래మ్의 생성

앞에서 언급했지만 본 논문에서는 홀로그래మ్을 생성하기 위하여 물체를 자체 발광하는 점들의 집합으로 간주하였다. 그리고 물체의 각 점들은 a_p 의 세기값, ϕ_p 의 위상값을 가지고 수직, 수평, 깊이 방향에서 x_p, y_p, z_p 에 위치한다고 가정한다. 이 때, n 번째 영상의 복소 진폭 $O_n(x, y)$ 는 식 (10)에서 나타난 것과 같다.

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

여기서 N_n 는 n 번째 프레임의 계산할 점들의 수를

나타내고, 파수(wave number) 벡터 k 는 $k = 2\pi/\lambda$ 로 정의되며 λ 는 자유공간상에서의 빛의 파장을 나타낸다. 편이상 $\exp(jax)$ 는 식 (10)에서 포함시키지 않았다.

본 논문에서는 홀로그래మ్을 생성하기 위하여 N-LT 기법을 사용한다. 이 N-LT 기법에서는 각 평면의 중앙에 있는 점에 해당하는 요소 프린지 패턴을 미리 만들어 록업테이블에 저장을 하고, 이를 불러와 x, y 방향으로 각각 이동시켜 평면상의 모든 점들을 표현하게 된다. 즉, 각각의 요소 프린지 패턴 $T(x, y; z_p)$ 는 임의의 깊이 z_p 에 대한 기준 세기의 프레넬 존 플레이트로 식 (8)과 같다.

제안하는 방법에서 첫 번째 프레임의 영상에 대해서는 기존의 N-LT 방식을 이용하여 홀로그래మ్을 생성한다. 그리고 첫 번째 프레임 영상과 생성된 홀로그래మ్ 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. 그리고 이전 프레임에서 차이에 해당하는 부분의 홀로그래మ్을 생성하여 버퍼에 저장된 홀로그래మ్ 패턴으로부터 빼주게 된다. 동시에 현재 프레임에서 차이에 해당하는 부분의 홀로그래మ్을 생성하여 버퍼에 저장된 홀로그래మ్ 패턴에 더해주어 현재 프레임에 해당하는 홀로그래మ్ 패턴을 완성하게 된다. 즉, 이전 프레임에서 사라진 부분에 대한 홀로그래మ్을 빼주고, 현재 프레임에서 새로 생긴 부분에 대한 홀로그래మ్을 더하여 현재 프레임의 홀로그래మ్을 생성하게 되고 다음과 같이 표현된다.

$$I_n(x, y) = I_{n-1}(x, y) - \sum_{p=1}^{N_d} a_{p_{n-1}} U_{n-1}(x - x_p, y - y_p; z_p) + \sum_{p=1}^{N_d} a_{p_n} U_n(x - x_p, y - y_p; z_p) \quad (11)$$

여기서, N_d 는 n 번째 프레임과 $n-1$ 번째 프레임 간 영상의 차이에 해당하는 점의 수를 나타낸다. 또한, $U_n(x, y; z_p)$ 는 n 번째 프레임의 PFP를 나타내고 다음과 같이 표현된다.

$$U_n(x, y; z_p) = \begin{cases} T(x, y; z_p) & \text{for changed part} \\ 0 & \text{for unchanged part} \end{cases} \quad (12)$$

마지막으로 생성된 홀로그래మ్ 패턴은 비디오 홀로그래మ్으로 출력되어 동시임시 버퍼에 저장되고 이러한 프로세스를 전체 동영상에 대하여 반복하여 3차원 동영상에 대한 비디오 홀로그래మ్을 생성

하게 된다.

4.2 프레임간의 차이 추출

본 논문에서는 실험 영상으로 200프레임의 320 × 240의 해상도를 갖는 3차원 동영상에 사용되었

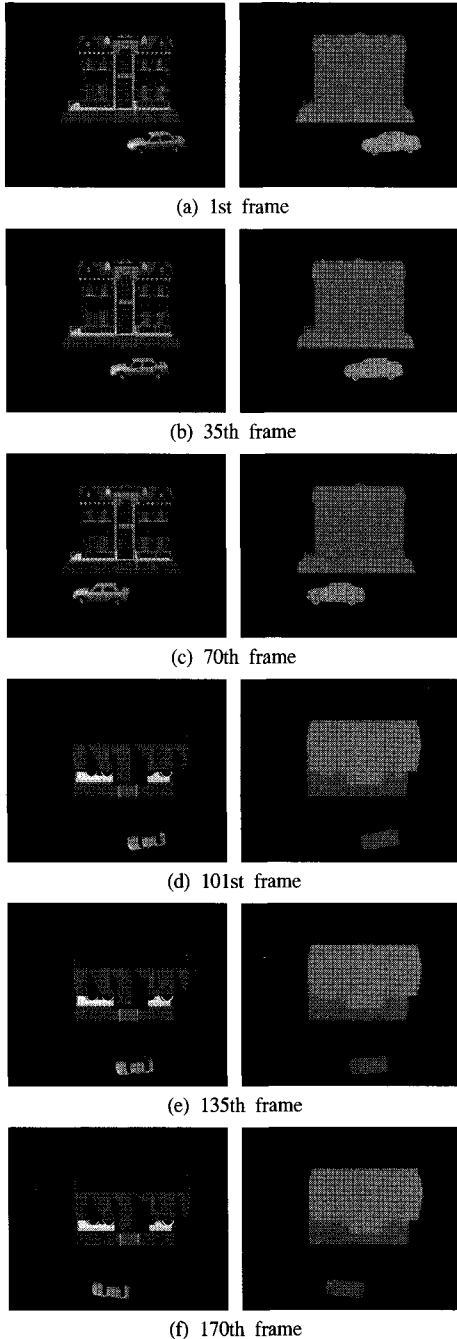


그림 5. 실험에 사용한 3차원 비디오 영상의 일부 프레임의 '밝기 영상'과 '깊이 영상'

다. 이 3차원 동영상은 두 부분으로 나뉘어 있는데 첫 번째 부분(Part I; 1 ~ 100 프레임)은 뒤에 있는 집 앞을 자동차가 오른쪽에서 왼쪽으로 이동하는 영상이고, 두 번째 부분(Part II; 101 ~ 200 프레임)은 첫 번째 부분을 위에서 본 모습의 영상이다. 각각의 영상에서 볼 수 있듯이 자동차는 정지해 있고 자동차는 이동하는 것을 볼 수 있다.

그림 5는 실험에 사용한 3차원 동영상의 일부 프레임의 밝기 영상과 깊이 영상을 나타낸다. 그리고 그림 6은 이전 프레임과 비교하여 차이가 있는 부분을 나타낸 영상으로 그림 6(a)를 보면 3차원 동영상에서 처음 입력되는 프레임이므로 영상 전체가 변화량이 된다. 그림 6(b),(c)의 경우는 집에 대한 영상 정보는 없고 자동차가 오른쪽에서 왼쪽으로 이동하는 듯한 영상을 보여준다. 하지만 그림 6(d)의 경우는 영상의 씬(scene)이 변하는 부분으로 집 주변을 이동하는 자동차의 정면에서 본 영상으로부터 위에서 본 영상으로 바뀌기 때문에 많은 부분이 바뀌게 된다. 그림 6(e), (f)의 경우를 보면 다시 집에 대한 영상 정보는 없고 자동차가 역시 오른쪽에서 왼쪽으로 이동하는 것을 볼 수 있다. 즉, 일반적인 3차원 동영상에서 첫 프레임 또는 씬이 바뀌는 부분을 제외하고, 각각의 씬 안에서는 영상의 일부 분만이 변하는 것을 볼 수 있다.

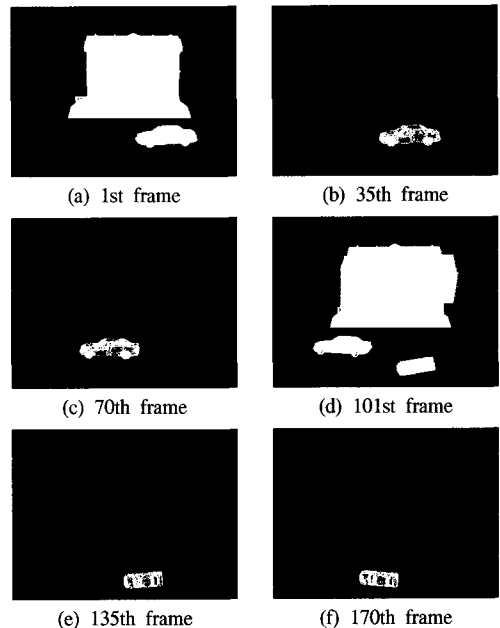


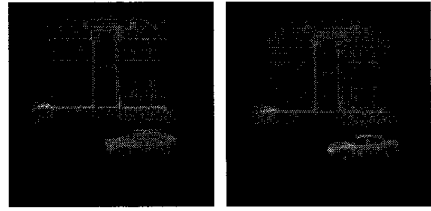
그림 6. 이전 프레임과 현재 프레임 간에 차이가 있는 부분의 영상

4.3 프레임간의 차이를 이용한 3차원 비디오 홀로그래ムの 생성

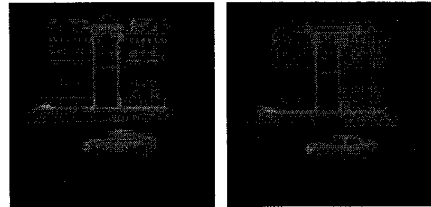
제안하는 방법에서는 한 프레임에 대하여 이전 프레임에서 사라진 부분에 대해 빼는 절차와 현재 프레임에서 추가된 부분에 대하여 더하는 절차의 두 가지 계산 절차가 필요로 하게 된다. 따라서 계산 시간이 두 배가 되기 때문에 제안하는 방법에서는 두 프레임간의 차이를 비교하여 차이가 50%가 넘는 경우에는 기존의 N-LT 방식을 이용하여 홀로그래ム을 생성한다. 하지만 이러한 경우는 3차원 동영상의 첫 번째 프레임이나 영상의 썸이 바뀌는 부분에 해당하고 그 이외의 대부분의 영역에서는 두 프레임간의 차이가 50% 미만이 된다. 따라서 3차원 동영상의 대부분의 프레임은 제안하는 방법에 의해 연산을 하게 된다.

본 논문에서는 앞에서 언급한 것처럼 그림 5의 3차원 비디오를 실험용 영상으로 사용하였다. 각 영상은 320×240 의 해상도를 갖고, 홀로그래ムの 크기는 $1,200 \times 1,200$ 으로, 홀로그래ムの 픽셀 크기는 $10 \mu m \times 10 \mu m$ 로 하였다. 또한 관찰자의 시정 거리는 200 mm 로 하였고, $200 \text{ mm} \times 0.003 = 60 \text{ microns}$ 이하의 간격의 점을 하나의 점으로 인식하게 됨으로 본 논문에서는 수직, 수평의 이산화 정도를 60 micron 으로 사용하였다. 따라서 이웃한 점을 표현하기 위해서는 $60 \mu m$ 를 이동하여야 하고 결국 6 픽셀을 이동하게 된다. 또한 이미지 공간의 첫 번째 포인트부터 마지막 포인트까지 표현하기 위해서는 요소 프린지 패턴을 수평/수직 방향으로 각각 $320 \times 6 \text{ 픽셀} = 1,920 \text{ 픽셀}$ / $240 \times 6 \text{ 픽셀} = 1,440 \text{ 픽셀}$ 만큼 각각 이동시켜야 한다. 다시 말하면, 홀로그래ム 패턴의 크기가 $1,200 \times 1,200$ 이므로, 전체 요소 프린지 패턴의 크기는 $3,120 (1,200 + 1,920) \times 2,640 (1,200 + 1,440)$ 의 크기를 사용하였다.

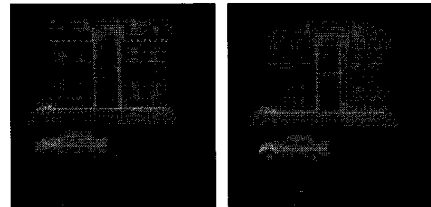
또한 각 프레임의 3차원 정보는 ‘밝기 영상’과 ‘깊이 영상’을 합성하여 구하고, 동일 깊이에 있는 물체 포인트들은 그 깊이를 대표하는 요소 프린지 패턴을 가지고 계산하기 위하여 깊이 방향으로 정렬하였다. 그리고 먼저 생성된 요소 프린지 패턴을 이용하여 기존의 N-LT 방식과 제안된 방법에 대하여 각각 홀로그래ム을 생성하였다. 그림 7은 기존 N-LT 방식과 제안된 방식에 의하여 생성된 홀로그래ム을 각각 복원한 영상의 일부 프레임을 나타낸다. 본 실험에서는 Intel CoreTM2 Quad CPU와 2.4 GHz 메모리로 구성된 PC와 Matlab 2009를 사용하여 수행했다.



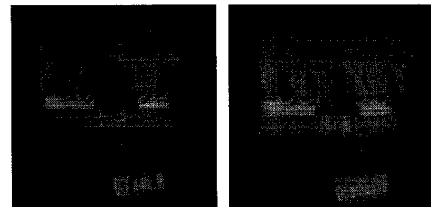
(a) 1st frame



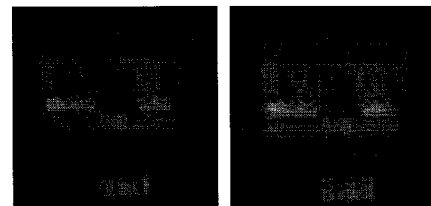
(b) 35th frame



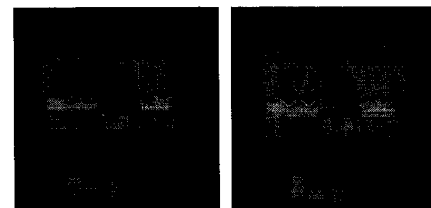
(c) 70th frame



(d) 101st frame



(e) 135th frame



(f) 170th frame

그림 7. 복원된 홀로그래ム 영상

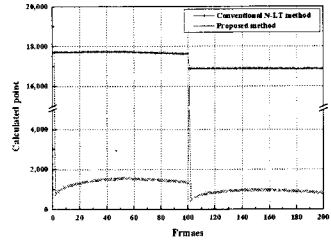
4.4 계산할 포인트 수와 계산 시간

그림 8(a)는 기존 N-LT 방식과 제안된 방식에

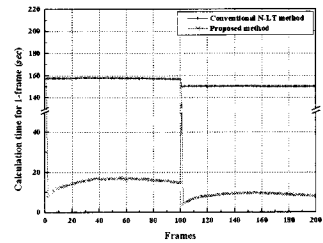
의한 테스트 3차원 동영상의 프레임당 계산 포인트 수를 나타낸다. 이를 보면 기존 N-LT 방식의 경우 프레임 내의 모든 포인트를 계산하기 때문에 각각의 Part 내에서 비슷한 계산 포인트 수를 갖는 것을 볼 수 있다. 반면 제안된 방식에서는 첫 번째 프레임과 101번째 프레임처럼 동영상의 첫 번째 영상이거나 씬이 바뀌는 부분의 경우는 기존의 N-LT 방식을 사용하기 때문에 기존 방식과 계산 포인트의 수가 같지만 그 이외의 부분에서는 자동차가 이동하는 부분에 대해서만 연산을 하므로 자동차의 속도에 따라 계산 포인트 수가 늘어나기는 하지만 기존 N-LT 방식에 비하면 상당히 줄어드는 것을 볼 수 있다. 그림 8(b)의 경우는 3차원 동영상의 각각의 프레임에 대해 홀로그램을 생성하는데 필요한 시간을 나타낸다. 이를 보면 그림 8(a)의 계산 포인트의 수와 마찬가지로의 모양을 보이고 있는 것을 볼 수 있다. 그림 8(c)의 경우는 각각의 프레임마다 한 포인트를 연산하는데 필요로 하는 평균 생성 시간을 나타낸다. 이를 보면 기존 N-LT 방식의 경우는 전 영역에 걸쳐 거의 같은 값들을 갖는 것을 볼 수 있다. 하지만 제안된 방법의 경우는 첫 번째 프레임과 101번째 프레임의 경우는 기존의 N-LT 방식을 사용하기 때문에 기존 방식과 생성 시간이 같게 나오지만 그 이외의 영역에서는 계산할 포인트의 수가 현저히 줄어들었기 때문에 포인트당 계산 시간이 현저히 줄어드는 것을 볼 수 있다.

표 1은 기존 N-LT 방식과 제안된 방식에 의한 테스트 3차원 동영상의 프레임당 평균 계산 포인트 수, 평균 계산 시간을 나타낸다. 표 1을 보면 기존의 N-LT 방식에서는 Part I, Part II 각각 17,702, 16,883 포인트의 연산을 하여야 한다. 하지만 제안하는 방법에서는 첫 번째 프레임과 101번째 프레임을 더하더라도 각각 1,555, 1,029 포인트를 연산하면 되어 연산량이 8.8%, 6.1%로 각각 줄어들게 된다. 또한 첫 번째 프레임과 101번째 프레임을 빼게 되면 1,392, 869포인트로 각각 7.9%, 5.1%로 줄어들게 된다.

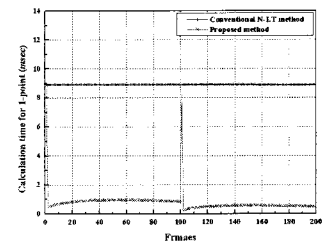
프레임당 평균 홀로그램 생성 시간 역시 기존의 N-LT 방식에서는 Part I, Part II 각각 157.483, 150.119초가 걸린다. 하지만 제안하는 방법에서는 첫 번째 프레임과 101번째 프레임을 더하더라도 각각 16.749, 9.967초가 걸려 계산 시간이 10.6%, 6.6%로 각각 줄어들게 된다. 또한 첫 번째 프레임과 101번째 프레임을 빼게 되면 15.333, 8.554초로 각각 9.7%, 5.7%로 줄어들게 된다.



(a) 계산 포인트 수



(b) 프레임당 계산 시간



(c) 포인트당 계산 시간

그림 8. 기존 N-LT 방법과 제안된 방법에 대한 비교

포인트당 평균 홀로그램 생성 시간 역시 기존의 N-LT 방식에서는 Part I, Part II 각각 8.897, 8.892 ms가 걸린다. 하지만 제안하는 방법에서는 첫 번째 프레임과 101번째 프레임을 더하더라도 각각 0.946, 0.590 ms가 걸려 계산 시간이 10.6%, 6.6%로 각각 줄어들게 된다. 또한 첫 번째 프레임과 101번째 프레임을 빼게 되면 0.866, 0.507 ms로 각각 9.7%,

표 1. 기존 N-LT 방법과 제안된 방법에 대한 비교

		기존 N-LT 방식	제안된 방식 (1/101프레임 포함)	제안된 방식 (1/101프레임 제외)
프레임당 평균 계산 포인트	Part I	17,702	1,555	1,392
	Part II	16,883	1,029	869
프레임당 평균 계산 시간 (sec)	Part I	157.483	16.749	15.333
	Part II	150.119	9.967	8.554
포인트당 평균 계산 시간 (ms)	Part I	8.897	0.946	0.866
	Part II	8.892	0.590	0.507

5.7%로 줄어들게 된다.

이러한 실험 결과로부터 3차원 동영상에 대한 홀로그래프 생성시 제안된 방법을 사용하여 동영상의 시간적 중복성을 제거하게 되면 기존 방법에 비하여 계산하여야 하는 포인트의 수가 현저하게 줄어들게 되어 결과적으로 홀로그래프를 고속으로 생성할 수 있게 된다.

V. 결 론

기존의 LT 방식에서는 가능한 물체영상의 모든 포인트에 대해 미리 계산 해놓은 요소 프린지 패턴을 이용함으로써 홀로그래프 합성시 엄청난 속도의 증가를 가져왔다. 그러나 이 방식의 가장 큰 단점은 미리 계산해놓은 요소 프린지 패턴의 양이 워낙 많고 따라서 이를 저장할 LT의 메모리 또한 엄청나게 증가하게 된다. 이러한 문제점을 해결하기 위해 제안된 N-LT 방식의 경우 계산 속도를 유지하며 필요로 하는 메모리 공간은 현저히 줄어들게 된다. 하지만 계산 속도가 현저히 줄어든다 하더라도 동영상의 경우는 초당 30프레임의 영상이 존재하기 때문에 계산해야 할 양이 많아지게 되어 많은 시간이 필요하게 된다. 따라서 본 논문에서는 기존의 N-LT 방식과 3차원 동영상의 시간적 중복성을 적용하여 3차원 동영상에 대한 홀로그래프를 고속으로 생성할 수 있는 방법을 제안하였다.

제안된 방법에서는 3차원 동영상의 시간적 중복성을 이용하여 계산해야 하는 포인트의 수를 현저하게 줄인 다음에 연산을 함으로써 계산 속도의 증가를 가져왔다. 3차원 동영상으로서 집 앞을 이동하는 자동차 영상을 사용한 실험결과는 제안된 방식에 비해 계산 포인트는 평균 7.5%로 줄어들게 되고, 계산 시간은 평균 8.6%가 줄어든 것을 볼 수 있다.

제안하는 방법은 입력되는 3차원 동영상의 특성을 이용하여 동영상 홀로그래프를 고속 생성 하게 된다. 즉, 입력되는 3차원 동영상의 특성이 본 논문에서 제안하는 방법과 맞지 않다면 효율은 떨어지게 될 것이다. 하지만 최악의 경우로서 카메라가 빠르게 이동하는 등의 경우에도 변화율이 50%가 넘어가게 되면 기존의 N-LT 방법을 이용하여 홀로그래프를 생성하게 되므로 최악의 경우 기존 방법과 같은 연산 시간을 갖게 된다. 하지만 일반적인 동영상에서는 씬이 바뀌는 부분이 아니면 이웃한 영상들은 비슷한 부분을 많이 갖게 된다. 따라서 본 논문에서

제안하는 방법을 이용한 효율적인 3차원 동영상 홀로그래프 생성이 가능하게 된다.

참 고 문 헌

- [1] 김은수, 이승현 공역, 3차원 영상의 기초, 기다리, 1998.
- [2] C. J. Kuo and M. H. Tsai, *Three-Dimensional Holographic Imaging* (John Wiley & Sons, 2002).
- [3] A. W. Lohmann and D. P. Paris, "Binary Fraunhofer holograms generated by computer," *Appl. Opt.* 6, 1739-1748 (1967).
- [4] M. Lucente, "Interactive computation of holograms using a look-up table," *J. Electron. Imag.* 2, 28-34 (1993).
- [5] 김승철, 김은수, "새로운 룩업테이블을 이용한 3차원 디지털 홀로그래프의 고속 합성 및 복원," *한국통신학회논문지*, 제33권 제3호 pp. 255-261, 2008. 03.
- [6] 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.*, 48 1030-1041 (2009).
- [7] P. Hariharan, *Optical Holography; Principles, techniques, and applications* (Cambridge Studies in Modern Optics, 1996).

김 승 철 (Seung-Cheol Kim)

정회원



2002년 2월 광운대학교 전자공학과 졸업

2004년 2월 광운대학교 전자공학과 석사

2007년 2월 광운대학교 전자공학과 박사

2007년 3월~현재 광운대학교 전자공학과 차세대 3D 디스플레이 연구센터 연구교수

<관심분야> 3차원 디스플레이, 홀로그래픽 디스플레이, 광공학

김 은 수 (Eun-Soo Kim)

중신회원

한국통신학회지 Vol.24, No. 9A 참조