

일반논문-11-16-2-07

# 다시점 3차원 비디오 재생 시스템 설계 및 구현

허영수<sup>a)</sup>, 박광훈<sup>a)‡</sup>

## Design and Implementation of Multi-View 3D Video Player

Young-Su Heo<sup>a)</sup> and Gwang-Hoon Park<sup>a)‡</sup>

### 요약

본 논문에서는 기존 비디오 재생 시스템보다 고속으로 동작하는 다시점 3차원 비디오 재생 시스템을 설계하여 구현하였다. 대용량의 다시점 영상 데이터를 고속으로 처리하기 위해 구성 모듈들을 병렬화하여 다중코어 프로세서 환경에서 최적의 속도를 얻을 수 있는 구조를 제안하였다. 병목지점의 병행성을 활용하기 위하여 복호화, 영상합성, 렌더링 모듈을 파이프라인 구조로 설계하였다. 부하 균형을 위하여 복호화 모듈을 시점 단위로 분할하고, 영상합성 모듈을 합성영상을 기준으로 기하적으로 데이터 분할하였다. 실험결과로서, 다시점 영상이 올바르게 합성되어 무안경식 다시점 입체디스플레이 상에서 시청 시 입체감을 느낄 수 있었으며, 제안하는 응용프로그램의 처리구조는 다중코어 프로세서를 최대 활용하여 대용량의 다시점 영상데이터를 고속으로 처리할 수 있었다.

### Abstract

This paper designs and implements a multi-view 3D video player system which is operated faster than existing video player systems. The structure for obtaining the near optimum speed in a multi-processor environment by parallelizing the component modules is proposed to process large volumes of multi-view image data at high speed. In order to use the concurrency of bottleneck, we designed image decoding, synthesis and rendering modules in a pipeline structure. For load balancing, the decoder module is divided into the unit of viewpoint, and the image synthesis module is geometrically divided based on synthesized images. As a result of this experiment, multi-view images were correctly synthesized and the 3D sense could be felt when watching the images on the multi-view autostereoscopic display. The proposed application processing structure could be used to process large volumes of multi-view image data at high speed, using the multi-processors to their maximum capacity.

Keyword: Multi-view 3D Video Player, Multi-view synthesizing

## 1. 서론

방송통신 기술과 영상 시스템 및 영상 압축 기술의 발전으로 인해 HDTV(High-definition TV)와 같은 고선명 방송 서비스가 제공되었으며, 이는 사용자의 요구사항에 따른 변화라 할 수 있다. 더 나아가 사용자는 오감을 총체적으로 자극하여 현실에서 경험하는 모든 것을 완벽하게 재현해주

a) 경희대학교 전자정보대학 전자정보학부 미디어랩  
Media Lab., College of Electronics and Information, Kyung Hee University

‡ 교신저자 : 박광훈 (ghpark@khu.ac.kr)

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음 (NIPA-2011-(C1090-1111-0001).

· 접수일(2010년7월2일), 수정일(1차:2011년1월10일,2차:1월17일), 게재확정일(2011년3월7일)

는 실감 미디어로의 발전을 기대하고 있다. 그 중 시각은 인간이 지각하는 신호 중 가장 큰 비중을 차지하며 현재 입체영상에 대한 관심과 수요가 급증하고 있다. 그것을 반영한 3차원 입체영화 ‘아바타’의 인기는 입체영상기술 활용을 통한 새로운 수익모델을 제시 하였으며, 영상콘텐츠 제작의 패러다임이 변화하고 있다는 것을 보여주었다. 의 과학 분야에서도 극한환경의 원격작업에서 현장감을 느끼도록 함으로써 작업효율을 높일 수 있어서 입체영상에 대한 수요가 증가하고 있다.

디스플레이 및 카메라 제작사들 또한 스테레오 카메라, 편광 렌티큘라 렌즈(Lenticular Lens)<sup>[1]</sup>, 셔터글라스<sup>[2]</sup>, 패럴랙스 배리어(Parallax Barrier)<sup>[3]</sup> 등의 방식과 같은 3차원 실감형 디스플레이 장치와 스테레오 카메라를 활발하게 출시하고 있다. 3차원디스플레이는 현재는 편광 선글라스 또는 셔터글라스 방식의 안경식 3차원 TV가 가정용으로 많이 보급되고 있다. 무안경 방식 중 하나인 무안경식 다시점 입체(Multi-view Autostereoscopic) 디스플레이는 안경식 3차원 디스플레이에 비해 아직 상용화 제품이 적지만 특수 안경을 착용하는 불편함 없이 입체영상을 시청할 수 있는 장점으로 인해 향후 TV나 영화관뿐만 아니라 모바일 영상기기에서도 많이 사용 될 전망이다.

이처럼 디스플레이 장치에 대한 수요가 2차원에서 3차원으로 이동해 감에 따라 비디오 촬영, 전송, 저장, 재생 시스템 또한 기존 시스템보다 더 발전 된 형태로 변모해 가고 있다. 그 중 재생 시스템에서의 변화란 두개 이상의 시점 영상을 동시에 복호화 하는 것, 입체 디스플레이 장치에 맞는 방식으로 두개의 영상을 하나의 영상으로 재구성(시점 분리방식)하거나 번갈아서 표시(시계열방식)하는 것이다. 현재 여러 종류의 입체 디스플레이들이 존재하지만 그중 실시간 처리 면에서 가장 문제가 되는 것은 무안경식 다시점 입체 디스플레이이다. 일반적으로 무안경식 다시점 입체 디스플레이는 두개의 시점을 사용하는 안경식 3차원디스플레이 보다 더 많은 수의 시점영상을 사용한다. 그에 따라 더 많아진 데이터양 때문에 발생하는 문제점으로 인해 기존과 다른 구조의 비디오 재생 시스템이 필요하다.

다시점 영상은 단일시점 영상의 수배에서 수십 배에 달하는 대용량 데이터이며 디스플레이 장치에서 재생하기 위

하여 복호화 후 복잡한 영상 매핑 과정이 필요하다. 2차원 영상 콘텐츠 재생을 목적으로 제작된 기존의 비디오 재생 시스템은 다시점 입체영상 재생에 최적화 되어있지 않기 때문에, 다시점 영상재생에서는 적정 프레임 처리속도를 만족시키지 못하는 경우가 생긴다. 과거에는 시간이 지남에 따라 프로세서의 클럭율이 높아짐으로 처리속도에 관한 문제가 해결되곤 하였다. 하지만 프로세서의 클럭율을 높일수록 전력소모율과 발열량은 기하급수로 증가하게 되어 프로세서의 클럭율 향상은 현재 한계에 도달한 상태이다. 이에 대한 대책으로 프로세서 제조사들이 택한 방법은 하나의 프로세서 안에 여러 개의 코어를 장착하여 병렬로 사용 가능하게 함으로써 프로세서의 처리속도를 증가시키는 것이었으며<sup>[4]</sup>, 현재 대표적인 프로세서 제작사인 Intel과 AMD사에서 출시되고 있는 상용 프로세서들의 코어 개수가 점점 늘어나고 있다.

과거에는 PC 및 영상기기에 내장되는 프로세서는 단일 코어였고 영상 재생 시스템 또한 단일 코어에 최적화 된 구조로 설계되었다. 그러나 현재 PC는 물론 모바일 영상기기에 듀얼코어 이상의 프로세서가 사용되고 있는 추세이다. 어플리케이션 또한 이러한 추세를 반영하여 변화된 시스템 환경을 최대한 활용하도록 설계해야 좋은 성능을 얻을 수 있다. 따라서 본 논문에서는 시스템을 병렬화 하여 복수개의 코어를 사용하는 프로세서를 활용하여 처리 속도를 높이는 방법을 사용하였다. 시스템을 다중코어에서 병렬로 가동하도록 하면 단일코어에 비해 대용량 데이터 처리속도를 높일 수 있을 뿐 아니라 더 낮은 전력을 소모하여 동일한 데이터 처리율을 얻을 수 있다. 본 논문에서는 대용량의 다시점 영상데이터의 고속 처리를 위해 소프트웨어 모듈들을 다중 코어 프로세서 환경에 적합하도록 병렬 구조로 설계 하고 응용프로그램을 구현하여 성능을 검증하였다. 다시점 영상 재생에 사용된 디스플레이는 V3i사의 8시점 무안경식 입체 디스플레이<sup>[5]</sup>이다.

본 논문의 구성은 다음과 같다. II장에서는 무안경식 다시점 입체 디스플레이 장치에 대하여 설명한다. III장에서는 다시점 영상의 합성에 대하여 설명한다. IV장에서는 II,III의 내용을 바탕으로 다시점 3차원 비디오 재생 시스템 구조 및 다중코어 프로세서 환경에서 최적의 속도를 얻기 위한

병렬화 방법을 제안한다. V장에서 구현한 응용프로그램의 성능을 평가한다. 마지막으로 VI장에서는 결론을 기술한다.

## II. 무안경식 다시점 입체 디스플레이 장치

디스플레이를 통해 눈으로 보는 것과 같은 현장감을 주려면 인간의 눈이 가지는 여러 가지 요소들을 구현해 주어야 한다. 2차원 디스플레이 장치를 통해 비디오를 시청 시 3차원 공간을 2차원 평면에 표시함으로 인하여 손실되는 인간의 시각 특성은 다음과 같다.

- 양안시차 - 두 눈에 다른 영상이 보임.
- 이동시차 - 사람이 머리를 움직이면 관측위치가 바뀌므로 인하여 다른 영상을 보게 됨.
- 시각조절 - 수정체의 모양을 변화시켜 관심물체를 향해 초

점을 맞춘다.

- 수렴 - 관심물체에 초점을 맞추기 위해 안구가 움직인다.

모든 3차원 디스플레이 장치들은 위 요소들 중 양안시차 요소를 기본적으로 지원한다. 그리고 나머지 요소들이 디스플레이 장치에 추가됨으로써 더욱 눈으로 보는 것에 가까운 효과를 낼 것이다. 양안시차를 이용한 입체영상 디스플레이 장치에 관한 기술은 다양한 방법들이 제안되어 왔는데 크게 안경식과 무안경식으로 분류 된다. 3차원영화 상영관 및 가정용 3차원TV로 보급되면서 잘 알려진 안경식 3차원 디스플레이는 특수한 안경을 착용하고 분리된 좌시점과 우시점 영상을 각 렌즈를 통하여 보는 방식을 사용한다. 무안경식 3차원 디스플레이는 특수 안경을 착용해야 하는 불편함 없이 3차원 영상을 제공하며 양안시차 요소와 이동시차 요소를 지원한다.

그림 1은 무안경식 다시점 입체 디스플레이의 원리를 보

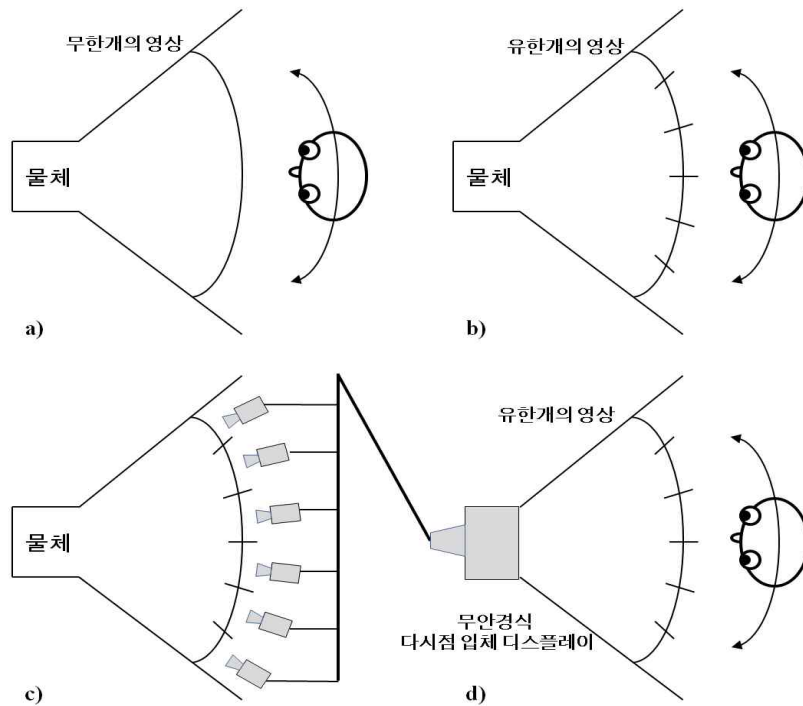


그림 1. 무안경식 다시점 입체 디스플레이 장치의 원리  
 Fig. 1. Principle of multi-view autostereoscopic display apparatus

여준다. 그림 1(a)는 관측자가 머리를 움직일 때 무한개의 서로 다른 영상을 보게 됨을 나타낸다. 그림 1(b)는 관측자의 머리가 이동할 수 있는 위치를 한정된 수로 제한하였을 때 한정된 수만큼의 영상을 볼 수 있음을 나타낸다. 따라서 그림 1(c)처럼 여러 개의 시점에서 촬영한 영상을 무안경식 다시점 입체 디스플레이를 사용하여 관측자의 머리위치가 바뀔때 따라 제한된 수의 서로 다른 영상을 보게 할 수 있다<sup>[6]</sup>. 무안경식 다시점 입체 디스플레이를 구현하는 방식에는 공간다중화방식, 다중영상방식, 시계열방식이 있으며 모든 무안경식 입체 디스플레이는 동일한 평면을 다른 시점에서 관측할 때 서로 다른 영상들이 보이게 하기 위해 기하 광학적 요소를 사용한다.

공간다중화 방식으로 무안경식 다시점 입체 디스플레이를 구현하는 방법에는 이차원 영상 패널에 일정한 거리를 두고 렌티큘러 렌즈<sup>[11]</sup>나 패럴렉스 배리어<sup>[12]</sup>를 설치하여, 관측자의 좌우 양안에 각기 다른 영상정보가 인식되도록 하여 입체감을 갖도록 하는 방법이다. 그림 2는 본 논문에서 시뮬레이션에 사용한 패럴렉스 배리어를 사용한 무안경식 다시점 입체 디스플레이 장치의 원리를 보여주며 2개의 시점을 사용하였을 때의 일례이다.

일반적인 패럴렉스 배리어를 이용한 무안경식 다시점 입

체 디스플레이 장치는 빛을 투과 또는 차단시키기 위한 가느다란 줄무늬 모양의 수직 슬릿(배리어)을 일정한 간격으로 배열시킨 다음 그 앞 또는 뒤에 적당한 간격을 두고 각 시점 영상을 교대로 배치하는 기술이다. 따라서 특정한 시점에서 이 슬릿을 통해 보면 기하광학 적으로 다시점 영상이 정확하게 분리되어서 입체감을 느끼게 된다. 즉, 모니터 화면 앞에 특수 안경 기능을 하는 줄무늬 모양의 패럴렉스 배리어를 설치해서 사용자가 안경을 쓰지 않고도 입체 영상을 인식하도록 하는 기술이다<sup>[13]</sup>.

### III. 다시점 영상 합성

본 장에서는 V3i사의 무안경식 다시점 입체 디스플레이 장치에서 사용하는 8시점 영상 합성 알고리즘에 대하여 설명한다<sup>[5]</sup>. 다시점 영상 합성은 N개의 시점으로 촬영된 영상을 매핑(Mapping) 알고리즘을 사용하여 디스플레이 프레임으로 사용할 하나의 영상으로 합성하는 것이다. 그림 3은 8개 시점의  $L \times M$  해상도의 영상을  $\frac{8}{3}L \times 3M$  해상도의 영상으로의 합성한 영상의 예를 보여주고 있다.

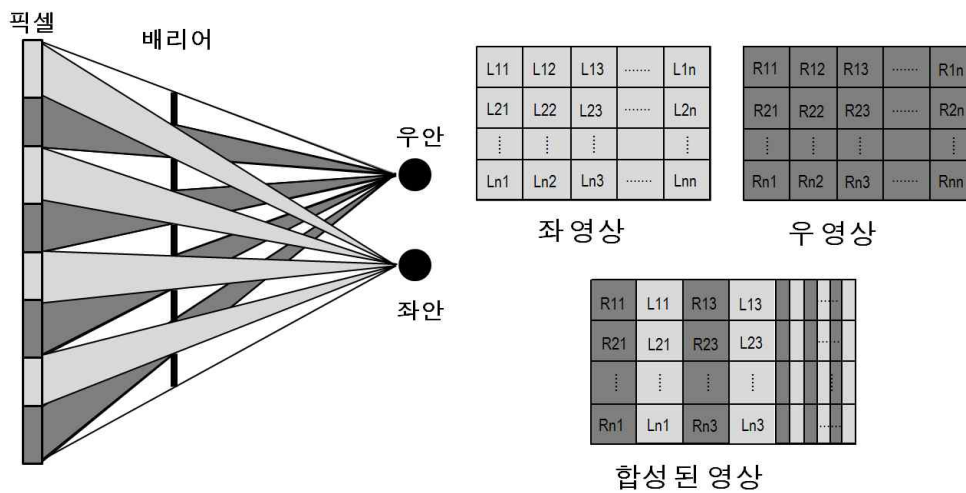


그림 2. 패럴렉스 배리어를 사용한 무안경식 다시점 입체 디스플레이 장치의 원리  
 Fig. 2. Principle of Multi-view autostereoscopic display apparatus using parallax barrier

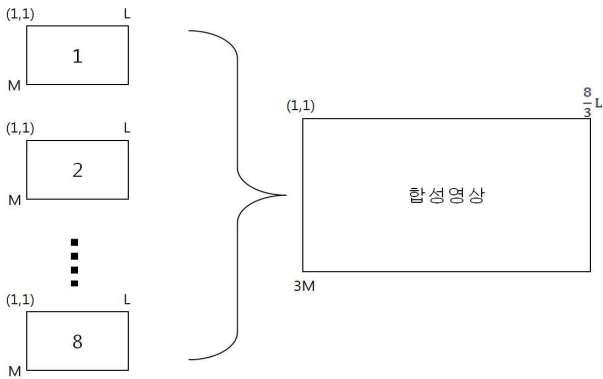


그림 3. 8시점 영상 합성 구조  
Fig. 3. The structure of 8-view image synthesizing

그림 4의 ‘A’는 디스플레이 프레임을  $j \times k$  다시점 화소 그룹으로 그룹화한 상태를 나타내며, ‘B’는 각 다시점 화소 그룹의 구성을 설명하기 위한 그림이다. 디스플레이 프레임은  $j \times k$  개의 다시점 화소 그룹으로 그룹화 되며, 이때  $j, k$ 는 자연수이다. 각 다시점 화소 그룹은 제1 화소부, 제2 화소부 및 제3 화소부로 이루어지며, 각 화소부는  $8 \times 1$ 개의 부화소(sub-pixel)로 이루어진다. 3개의 부화소는 하나의 화소를 구성한다.

그림 5는 V3i사의 무안경식 다시점 입체 디스플레이 장치에서 적용한 8시점 영상 매핑 방법을 도시하고 있다<sup>[5]</sup>. 상단의 번호는 화소번호이다. 좌측의 번호는 화소부의 번

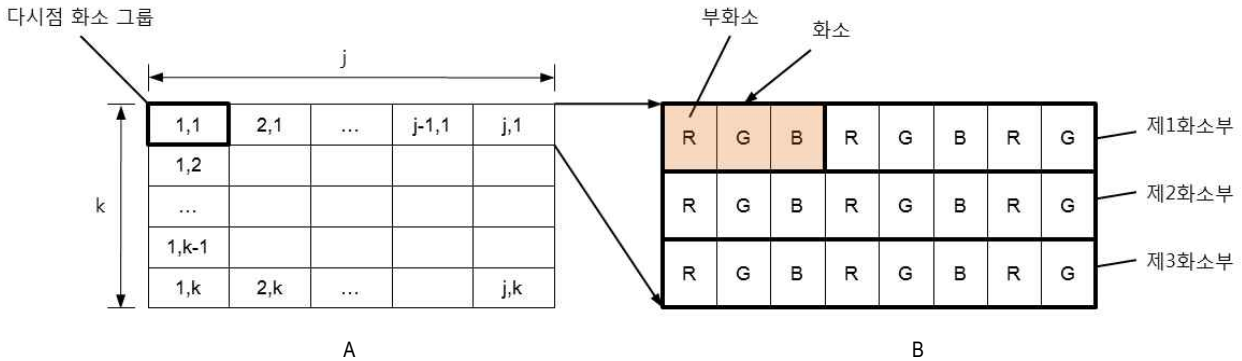


그림 4. 디스플레이 프레임의 다시점 화소그룹 분할  
Fig. 4. Partition the display frame into multi-view pixel group

		1		2		3		4		5		6		7		8			
No		R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1		1.1R	8.1G	7.1B	6.1R	5.1G	4.1B	3.1R	2.1G	1.1B	8.2R	7.2G	6.2B	5.2R	4.2G	3.2B	2.2R	1.2G	8.3B
2	1	8.1R	7.1G	6.1B	5.1R	4.1G	3.1B	2.1R	1.1G	8.2B	7.2R	6.2G	5.2B	4.2R	3.2G	2.2B	1.2R	8.3G	7.3B
3		8.1R	7.1G	6.1B	5.1R	4.1G	3.1B	2.1R	1.1G	8.2B	7.2R	6.2G	5.2B	4.2R	3.2G	2.2B	1.2R	8.3G	7.3B
4		7.1R	6.1G	5.1B	4.1R	3.1G	2.1B	1.1R	8.1G	7.2B	6.2R	5.2G	4.2B	3.2R	2.2G	1.2B	8.2R	7.3G	6.3B
5	2	6.1R	5.1G	4.1B	3.1R	2.1G	1.1B	8.1R	7.1G	6.2B	5.2R	4.2G	3.2B	2.2R	1.2G	8.2B	7.2R	6.3G	5.3B
6		6.1R	5.1G	4.1B	3.1R	2.1G	1.1B	8.1R	7.1G	6.2B	5.2R	4.2G	3.2B	2.2R	1.2G	8.2B	7.2R	6.3G	5.3B

그림 5. 8시점 영상의 매핑 방법  
Fig. 5. Method of 8-view image mapping

호를 의미하며, 그 오른쪽의 숫자는 다시점 화소그룹의 행 번호를 의미한다. 임의의 열에 배치된 k개의 다시점 화소그룹의 m행의 1 화소부를 미리 정해진 시점 배열 순서대로 합성 영상을 매핑 시킨다. m행의 2 화소부는 m행의 1 화소부의 시점 배열순서는 유지하되, 1 부화소만큼 좌측으로 이동시켜 합성 영상을 매핑 시키며, m행의 3 화소부는 m행의 2 화소부와 동일하게 합성영상을 매핑 시킨다. 또한, m행과 인접하여 배치되는 m+1행의 1 화소부는 m행의 3 화소부의 시점 배열순서는 유지하되, 1 부화소만큼 좌측으로 이동시켜 합성 영상을 매핑 시킨다. 그리고 m+1행의 2 화소부는 m+1행의 1 화소부의 시점 배열순서는 유지하되, 1 부화소만큼 좌측으로 이동시켜 합성 영상을 매핑 시키며, m+1행의 3 화소부는 m+1행의 2 화소부와 동일하게 합성 영상을 매핑 시킨다.

예를 들면, 8개의 영상 중 첫 번째 영상의 1.1번째 화소의 ‘Red’ 성분을 배열하고 8.1번째 영상의 첫 번째 화소의 ‘Green’ 성분을 배열하고, 7.1번째 영상의 첫 번째 화소의 ‘Blue’ 성분을 배열하는 순으로 진행이 되며 8개의 부화소 배열이 완료되면 두 번째 1.2 번째 화소의 ‘Red’ 성분이 배

열되고 8.2번째 화소의 ‘Green’ 성분이 배열되고 7.2번째 부화소의 성분이 배열된다. 8개의 부화소의 배열이 완료되면 영상들의 2번째 화소를 1차 배열과 같이 배열을 하여 8개의 부화소를 진행하는 순으로 매핑 한다. 매핑의 2번째 줄은 1번에 사용되었던 화소의 다른 색들의 혼합이 된다.

이와 같이 합성 영상을 매핑하면, 그림 5와 같이 각 시점 별 합성영상은 R, G, B 부화소가 모여서 화소를 이루지 않고, R, G, B 부화소 중 2개의 부화소만으로 화소를 구성하게 된다. 즉, (1, 1)은 R 부화소와 2개의 G 부화소로 구성되며, (2, 1)은 B 부화소와 2개의 G 부화소로 구성된다. 따라서 그림2에서와 같이 8개 시점의  $L \times M$  해상도 영상들을 합성한 영상의 해상도는  $\frac{8}{3}L \times 3M$ 이 된다.

그림 6은 그림 5의 8시점 영상 매핑 중 첫 번째 행의 RGB 화소들이 배리어를 통해 관측자에게 보여지는 과정이다. 패널상의 화소들은 R, G, B로 표시된 필터를 통해서만 관측자에게 보여지며 S로 표시된 부분은 빛을 차단한다. 따라서 그림 6의 예에서는 좌안에는 4번 영상이, 우안에는 7번 영상이 보여지게 되어 입체 영상을 관찰 할 수 있게 된다. 만

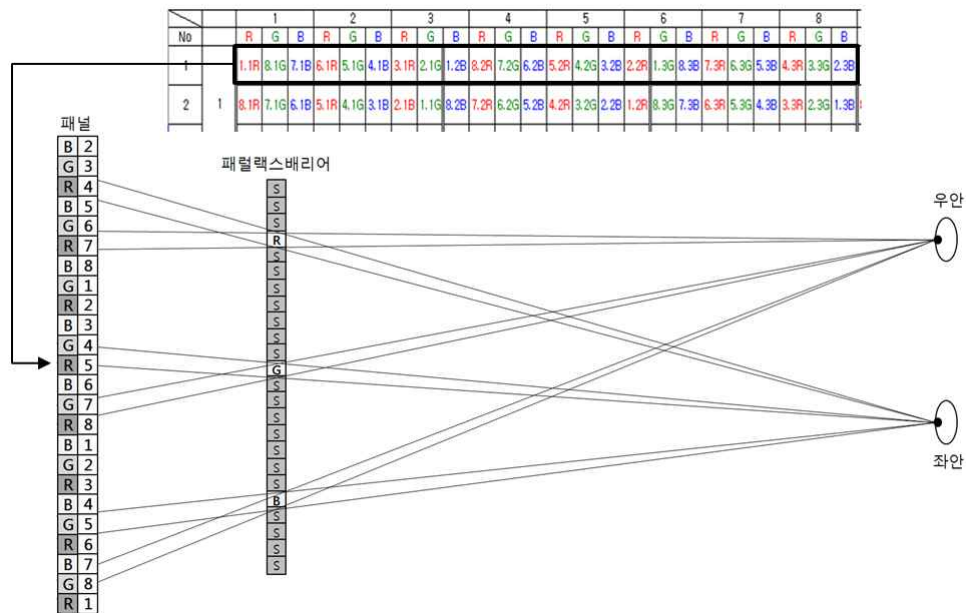


그림 6. 8시점 영상의 매핑 방법  
Fig. 6. Method of 8-view image mapping

일 관측자가 좌측으로 한 시점만큼 이동하면 좌안에는 5번, 우안에는 6번 영상이 관찰되게 된다. 이와 같은 원리로 관측자는 위치에 따라 다수의 시점을 관측할 수 있다<sup>7)</sup>.

#### IV. 다시점 3차원 비디오 재생 시스템의 구현

본 장에서는 II, III 장에서 소개한 내용을 바탕으로 다시점 3차원 비디오 재생을 위한 시스템 구조와, S/W 응용프로그램 구현 시 다중코어 프로세서 환경에서 최적의 속도를 얻을 수 있는 병렬화 방법을 제안한다.

본 논문에서 구현한 다시점 3차원 비디오 재생 시스템은 최종사용자에게는 다시점 비디오를 시청할 수 있도록 하고, 개발자에게는 임의의 다시점 합성 알고리즘을 적용하고 시험 할 수 있는 환경을 제공한다. 또한 대화형 인터페이스를 지원하여 2차원/3차원 모드 변환과 2차원 모드 재생 중 시점 전환이 가능하게 하였다. 제안된 시스템의 입력데이터는 압축된 다시점 영상 시퀀스이며, 시점 개수를 추가적으로

로 입력 받는다.

그림 7은 다시점 3차원 비디오 재생 시스템의 동작과정을 도시하고 있다. 복호화기를 거쳐서 나온 재생영상을, 시점발생기에서 다시점 비디오 합성과정(3차원 모드 선택 시)을 거치거나, 또는 사용자가 선택한 시점의 재생화면을 렌더링 하여 (2차원 모드 선택 시) 무안경식 다시점 입체 디스플레이 또는 2차원 디스플레이에서 재생한다. 이때 재생모드 관리는 재생 제어기에서 수행하는데, 프레임 위치를 선택하거나 (프레임 선택기), 다시점 3차원/2차원 모드를 선택하거나 (2차원/3차원 모드 선택기) 또는 시청하는 시점을 선택(시점 선택기)을 한다.

그림 8은 다시점 3차원 비디오 재생과정의 데이터 흐름, 즉 복호화, 영상합성, 렌더링을 거치는 과정을 도시하고 있다. 입력되는 다시점 비디오 비트스트림은 동일 시점(時點)에서 촬영된 8개 시점(視點) 영상을 압축한 부호화된 비트스트림이다. 이는 복호화기를 거쳐 8개의 재생영상으로 복원되어 메모리에 저장된다. 복원된 영상은 III장에서 설명된 영상합성 모듈에서 매핑 알고리즘을 통해 합성영상으로 변

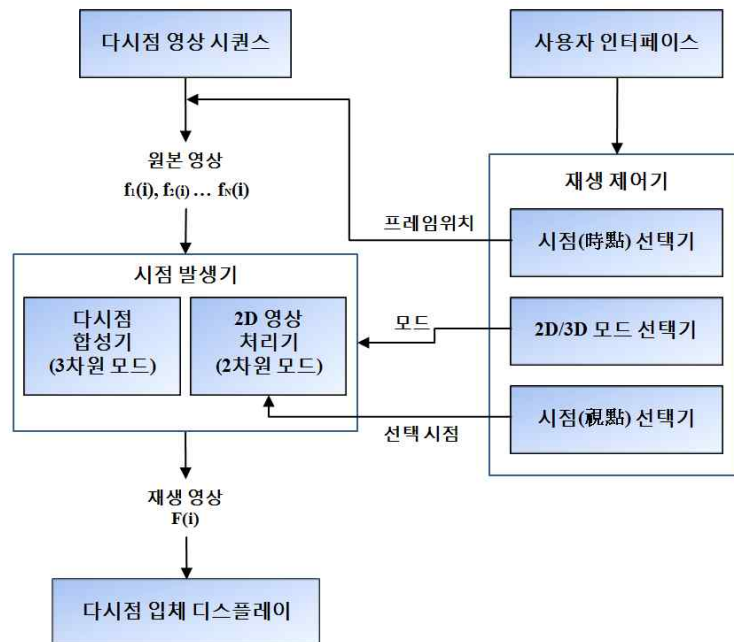


그림 7. 다시점 3차원 비디오 재생 시스템의 동작 과정  
 Fig. 7. Operating process of Multi-view 3D video player system

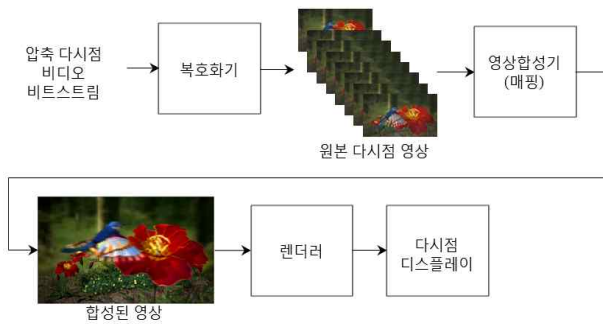


그림 8. 다시점 3차원 비디오 재생 과정의 데이터 흐름  
 Fig. 8. Data flow of the Multi-view 3D video player

환된다. 이때, 영상합성 모듈을 거쳐서 나온 합성영상을 확대 또는 축소하면 화소 매핑 구조가 깨지므로, 필요하다면 매핑 알고리즘이 적용되기 이전에 미리 디스플레이 장치 해상도에 맞게 해상도를 수정해야 한다. 합성된 영상은 렌더러를 통해 화면버퍼에 저장되어 무안경식 다시점 입체 디스플레이 장치에 표시되며, 매 프레임마다 이 과정들이 반복된다.

제안된 시스템은, 다중코어 프로세서를 최대한 활용하기 위하여 병렬구조로 설계하였다. 먼저, 프로세서가 많은 처리시간을 소모하게 하는 병목지점을 알기 위하여 각 작업의 복잡도가 전체에서 차지하는 비중을 분석하였다. 분석에 사용된 입력은 720x360 해상도의 H.264/AVC High Profile로 압축된 다시점 비디오 시퀀스이다.

표 1은 인텔 병렬 증폭기(Intel Parallel Amplifier)<sup>[8]</sup>로 CPU 클럭 소모량을 기준으로 측정한 다시점 3차원비디오 재생기의 주요 모듈에 대한 복잡도 분석 결과이다. 첫 번째 열은 모듈이 담당하는 작업이고, 두 번째 열은 전체 작업

표 1. 주요 모듈의 복잡도 분석.  
 Table 1. Analysis of complexity of main modules.

작업부하	복잡도 비중(%)
복호화	55.9
영상합성	25.1
렌더링	15.5
합계	96.5

수행을 100%로 보았을 때, 해당 모듈의 복잡도 비중이다.

표 1에 도시한 바와 같이, 세 개 주요 모듈의 비중이 전체에서 차지하는 비중이 96.5%일 경우, 암달의 법칙(Amdahl's law)<sup>[9]</sup>에 의하여 병렬화로 얻을 수 있는 이론적인 한계속도 향상률은, 8개의 코어 사용 시  $1/(0.035+0.965/8) = 6.41$ (배)이다. 표 1의 분석을 통하여, 복호화와 영상합성 모듈은 작업 단위 병렬성을 부여할 만한 비중을 가지고 있음으로 판단할 수 있다. 렌더러 또한 복잡도 비중이 다소 작더라도 다른 모듈들과 독립적으로 병행수행하는 것이 효율적이다.

특히, 렌더링은 영상 콘텐츠 초당 프레임 수에 따라 주기적인 처리 마감시간이 지켜져야 하기 때문에 렌더링 모듈과 바로 전 모듈 사이에 버퍼를 두고 병행 수행함으로써 이전 과정들의 처리시간 변동성을 관리해 주어야만, 화면 재생 시 초당 프레임수를 일정하게 유지할 수 있다.

작업 단위 병렬성을 부여했을 때 다음과 같은 두 가지 문제점이 발생한다.

첫 번째 문제는 각 모듈 간에 입출력 의존성이 존재한다는 것이다. 복호화 모듈의 출력이 영상합성 모듈의 입력이 되고 영상합성 모듈의 출력이 렌더링 모듈의 입력이 되는데, 본문에서는 이렇게 입출력관계로 순차적으로 진행되는 모듈들을 병렬화 하기 위하여 파이프라이닝(pipelining)을 사용하였다.

그림 9는 다시점 3차원비디오 시스템의 파이프라인 병렬 처리 구조를 보여주고 있다. 그림에서,  $Src_n$ ,  $Dst_n$ ,  $Out_n$ 은 각각 재생영상, 합성영상, 출력영상이며  $Level_n$ 에 각 단계(복호화, 영상합성, 렌더링)의 출력큐 버퍼의 내용을 도시하였다. 3 단계의 파이프라인 처리는 3번째 단위시간부터 전체 단계가 가동하며, 이를 위하여 각 모듈 당 출력버퍼의 최소 용량은 Level 1인 복호화 모듈에서 '2', Level 2인 영상합성 모듈에서 '1'이다. 이는 의존성 해결을 위하여 추가적인 메모리 사용을 감수해야 함을 의미한다.

두 번째 문제점은, 첫 번째 문제를 해결하기 위하여 파이프라이닝을 사용함으로써 발생된다. 각 모듈이 부담하는 작업부하의 차이로 인하여 입력데이터가 없는 상태의 단계에서 정지현상이 발생하는 부하 균형(load balancing) 문제가 발생한다. 작업 단위로 분산시킨 스레드(Thread)들의 각 작



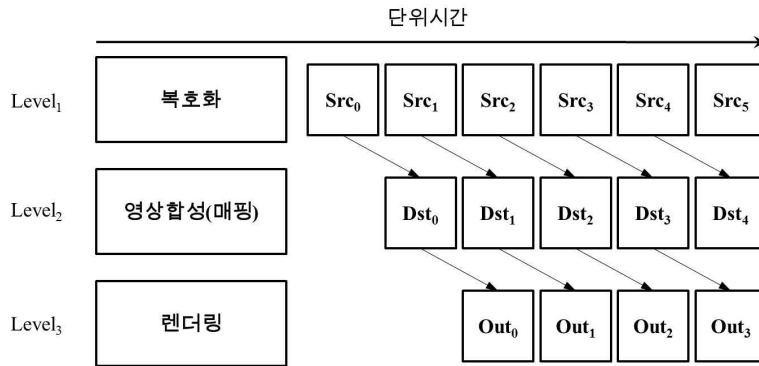


그림 9. 비디오재생기의 파이프라인  
Fig. 9. Pipeline of Video Player

업량이 다수의 프로세서에 고르게 분배되지 않을 경우 부하 균형 문제가 발생하는데 이는 반드시 피해야 하는 상황이다. 운영체제의 스케줄링은 일반적으로 처리량(Throughput)이나 응답시간(Response time)의 향상을 목표로 하는데, 이를 위하여 작업부하의 불균형을 용납하게 된다. 따라서 응용프로그램의 속도를 향상시키기 위한 방법으로 병렬화를 선택하였을 때에는 구현 단계에서의 세밀한 처리가 필요하다<sup>[10]</sup>.  
본 논문에서는, 파이프라인의 부하 균형 문제를 해결하기 위한 효과적인 방법으로서 그림 10에 도시한 바와

같이 Master-worker 모델<sup>[11]</sup>을 사용하여 데이터 단위 병렬성을 부여하였다. Master-worker 모델의 처리과정은 다음과 같다.

- 1) Master는 전처리 후, 분할한 작업을 작업풀(bag of tasks)에 저장하고 N개의 worker를 가동시킨다.
- 2) Master는 작업풀에 쌓인 작업들이 모두 처리 될 때까지 유휴상태인 worker에 작업을 할당한다.
- 3) 작업이 완료되고 모든 worker가 종료되면 master는 후처리를 수행하고 종료한다.

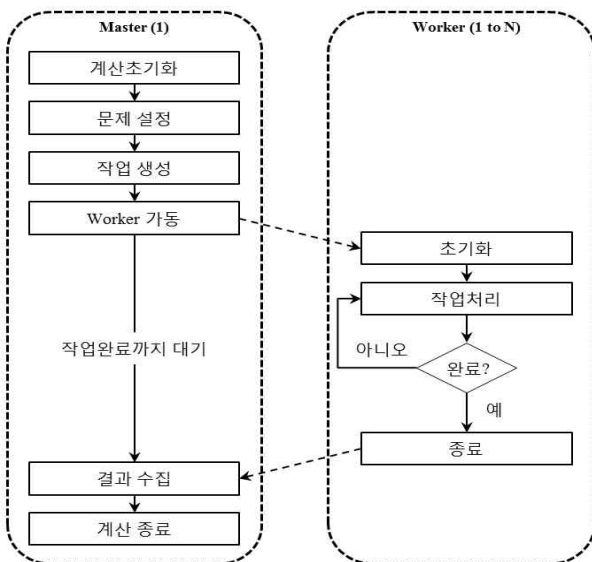


그림 10. Master-worker 모델 구조  
Fig. 10. Master-worker model structure

Master-worker 모델은 확장성이 매우 좋아서 분할된 작업들의 로드가 대체적으로 비슷하다면, 작업개수와 스레드개수에 상관없이 자동적으로 부하 균형을 맞춘다는 장점이 있어서 복호화와 영상합성의 데이터 단위 병렬화에 적합하다.

본 논문에서는 데이터 단위 병렬성을 부여할 모듈을 Master로, N개 스레드로 구성되어 N개의 worker의 역할을 하는 스레드풀을 사용함으로써 master-worker 모델을 구현하였다. 스레드들은 스레드풀이 담당하는 모듈(Master)의 작업풀에 정의된 작업들을 처리하는 것을 반복함으로써 worker의 기능을 수행한다. 예를 들어, n개의 압축 비디오 비트스트림은 n개의 데이터가 존재하는 작업풀(bag of tasks)이 되고, Master는 작업풀 내의 데이터들을 유휴상태의 스레드(worker)에 할당하여 처리하는 방식으로 구현하

였다.

초당 수십 프레임을 처리해야 하는 상황에서 매 프레임마다 스레드를 새로 생성하여 병렬처리한 후 스레드를 종료시키는 방법을 사용할 시에는 스레드 생성과 종료에 대한 처리시간으로 인하여 오버헤드가 크다. 따라서 재생 정지요청이 있지 않는 한 생성된 스레드들을 유지하면서 프

레이름을 연속으로 처리하는 스레드풀 방식을 사용하는 것이 적절하다<sup>[12]</sup>.

그림 11은 8개 시점 영상이 압축된 다시점 비디오 비트스트림을 4(N=4)개의 스레드에서 데이터 단위 병렬처리 하는 방법의 일례이다. 8개의 각 시점영상을 분할된 데이터로 간주함으로써 8개의 데이터를 N(8≥N)개의 스레드가 처리

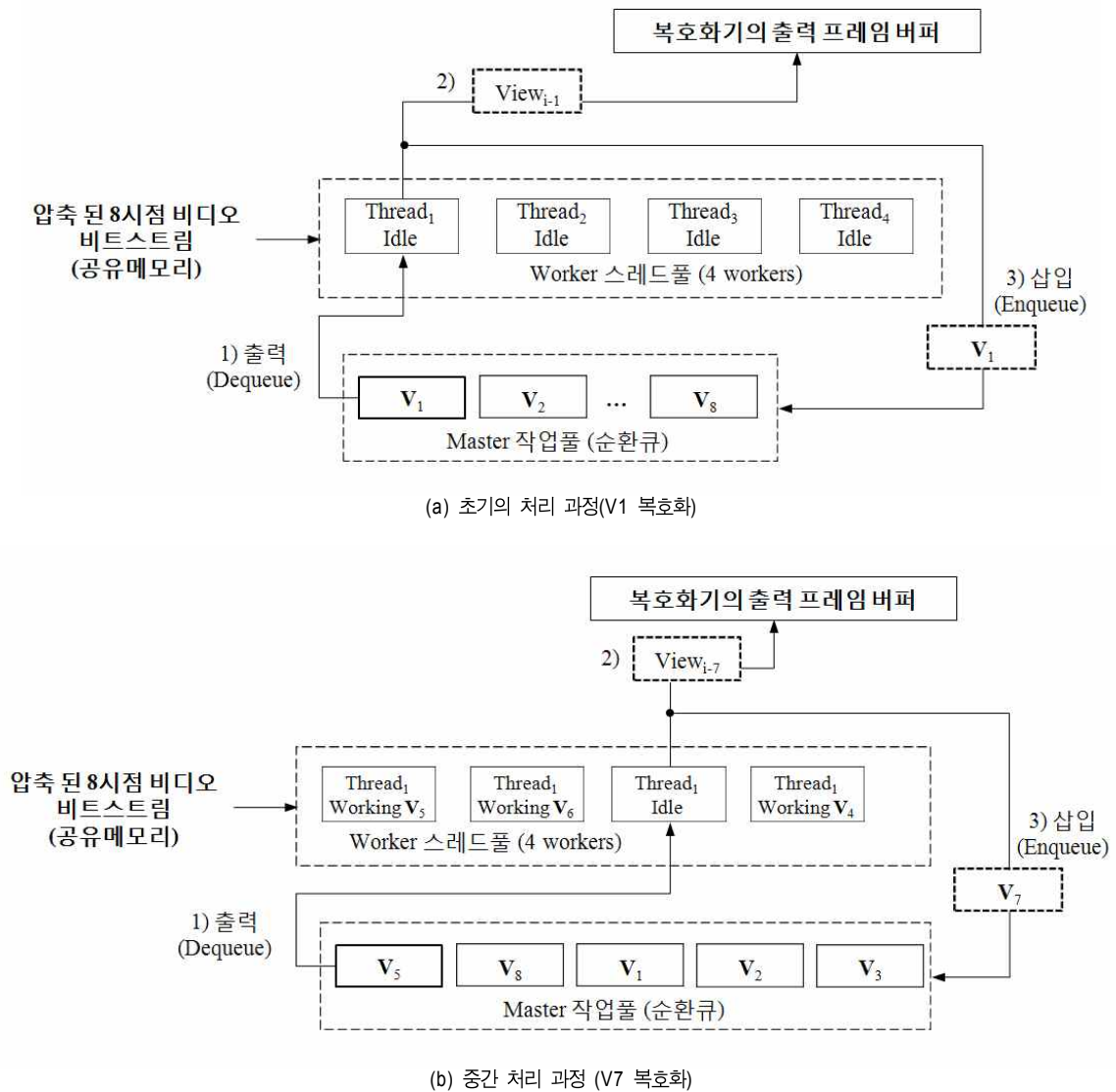


그림 11. 복호화 모듈의 데이터분할  
Fig. 11. Data decomposition of decoding module

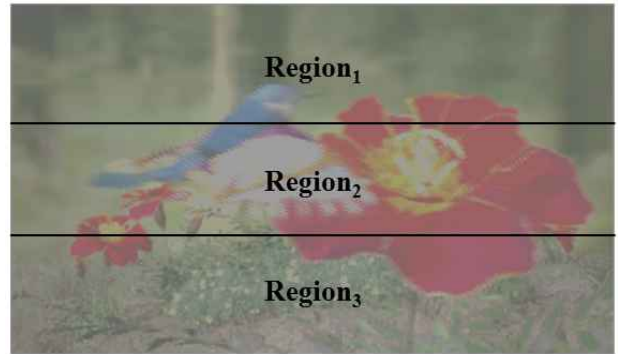
한다. 각각의 시점 영상은 공유메모리(Shared Memory) 방식으로 스레드들이 공유하고, 동시에 하나의 스레드만이 접근하도록 동기화 하였다. Master 작업풀은 순환큐(Circular Queue) 구조이며 초기에는 스레드가 처리할 시점 번호를 의미하는 신호( $V_1 \sim V_8$ )가 저장되어 있다. 각 스레드가 복호화 한 프레임은 파이프라인의 복호화단계 출력큐로 저장된다.

그림 11(a)와 (b)에 각각 초기 및 중간 처리과정을 도시하였다. 그림에 표기한 "1)"~"3)"은 처리 순서를 의미하며 자세한 동작과정은 다음과 같다.

- 1) Master는 유휴상태의 스레드가 있는지 확인한 후 작업 풀 순환큐를 출력(Dequeue)하여 나온 시점 번호  $V_j$ 를 스레드에 할당한다. 그림 11(a)에서는 모든 스레드가 유휴상태이고 작업풀 순환큐가 초기상태인 경우이다. Master는 유휴상태의 스레드 중 하나를 임의로 선택하여(위 예에서는 Thread<sub>1</sub>을 선택하였다.) 작업풀의 데이터  $V_1$ 를 할당한다. 그림 11(b)에서는 Thread<sub>3</sub>을 제외한 모든 스레드가 데이터를 처리하는 중이다. Master는 Thread<sub>3</sub>에 작업풀의 데이터  $V_7$ 를 할당한다.
- 2) 시점 번호를 얻은 스레드는  $i$ 번째 프레임의  $j$ 번 시점 영상인  $View_{i,j}$ 를 복호화 한다. 스레드는 복호화 된  $View_{i,j}$ 를 디코더의 출력버퍼에 저장한다.
- 3) 작업을 완료 한 스레드는 시점 번호  $V_j$ 를 Master의 순환큐에 삽입(Enqueue)한 후 유휴상태가 되고 Master의 다음 호출을 기다린다.

영상합성 모듈의 입력 데이터인 재생영상은 기하학적 분할방식<sup>[10]</sup>으로 데이터 단위 병렬성을 부여할 수 있는데, 화소매핑이 될 합성영상을 수직 방향으로  $M$ 개로 분할한 부영역들을 분할된 데이터로 간주한다. 그림 12는 합성영상을 3( $M=3$ )분할 한 일례로, 분할영역인 Region<sub>1</sub>~Region<sub>3</sub>은 Master의 작업풀 순환큐의 데이터로서 초기화된다.  $N(M \geq N)$ 개의 스레드로 구성 된 worker는 데이터가 가리키는 영역에 매핑을 수행한다. 복호화 된  $i$ 번째의 8개 시점 영상들인  $View_{i,1} \sim View_{i,8}$ 은 스레드들의 입력데이터로서 공유메모리(Shared memory) 방식으로 공유되며 복수개의 스레드가

동시에 자유롭게 접근 가능하다.



합성영상이 저장 될 프레임

그림 12. 합성영상의 기하학적 분할의 일례

Fig. 12. An example of geometric decomposition of synthetic image

그림 13은 두 개의 스레드에서 출력영상을 3 분할하여 처리할 때의 일례이다 ( $N=2, M=3$ ). 데이터 풀은 출력영상의 분할된 영역을 의미하며, 처리과정은 복호화 모듈의 데이터 분할 병렬처리 방법과 유사하다. Worker의 출력인 합성영상은 파이프라인의 영상합성단계 출력큐에 쌓이게 된다.

그림 13에 도시한 "1)"~"3)"은 처리순서를 의미하며 자세한 내용은 다음과 같다.

- 1) Master는 유휴상태의 스레드가 있는지 확인한 후 작업 풀 순환큐를 출력하여 나온 영역 인덱스  $R_j$ 를 스레드에 할당한다. 그림 13에서는 모든 스레드가 유휴상태이고 작업풀 순환큐가 초기상태인 경우이다. Master는 유휴상태의 스레드 중 하나를 임의로 선택하여(위 예에서는 Thread<sub>1</sub>을 선택하였다.) 작업풀의 데이터  $R_1$ 를 할당한다.
- 2) 영역 인덱스를 얻은 스레드는 공유메모리에 저장되어 있는 원본영상  $View_{i,1} \sim View_{i,8}$ 과 영상매핑 알고리즘을 사용하여  $j$ 번 영역에 매핑을 수행한다.
- 3) 작업을 완료 한 스레드는 영역 인덱스  $R_j$ 를 Master의 순환큐에 삽입한 후 유휴상태가 되고 Master의 다음 호출을 기다린다.

영상합성 모듈의 출력큐에 저장된 합성영상은 렌더링 모

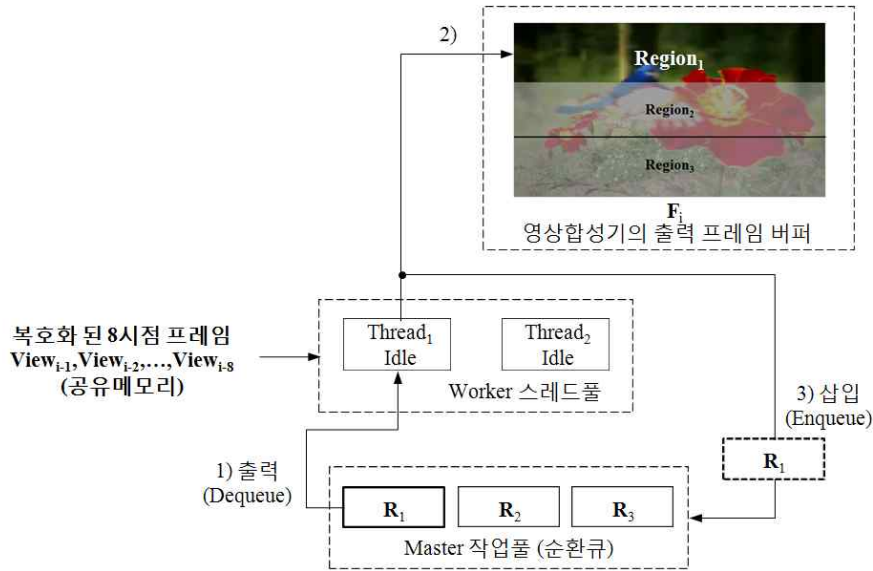


그림 13. 영상합성 모듈의 데이터분할  
Fig. 13. Data decomposition of image processing module

들에 연결 되어 화면에 디스플레이 된다. 렌더링 모듈은 일반적으로 하드웨어에 의해 고속으로 처리되므로 데이터 단위 병렬화를 고려하지 않았다. 전체 작업의 처리 속도가 가장 느린 작업의 속도에 동기화 된다는 것을 바탕으로, 부하 균형을 위한 데이터 단위 병렬화 과정에서 각 모듈에 배정할 Worker 스레드 수를 결정하는 기준을 다음과 같이 정할 수 있다.

- 1) 동시에 수행되는 N개의 작업들에 같은 수준의 속도향상을 부여하더라도 전체속도는 가장 느린 작업의 속도가 된다.
- 2) 동시에 수행되는 N개의 작업들에 속도향상을 부여할 때, 각 작업에 대한 속도향상 부여의 비율이  $\frac{P_i}{P_{Total}}$  일 때 최적이다. 이때  $P_i$  = 전체작업 중 작업 P의 비중이며,  $P_{Total} = \sum_{i=1}^N P_i$ 이다.
- 3) 동시에 수행되는 N개의 작업들에 속도향상을 부여할 때, 최적 비율을 초과하는 속도향상 부여는 초과된 만큼의 오버헤드가 된다.

예를 들어 시스템의 한계성능을 끌어낼 수 있는 최적의 스레드 수가 8이고, 표 2에서와 같이 세 개의 작업의 복잡도 비중이 55:25:15 라면 이 비례를 기준으로 각 작업에 5:2:1과 같이 대략적인 비율로 스레드 수를 배정함으로써 데이터 단위 병렬화 강도를 조절한다.

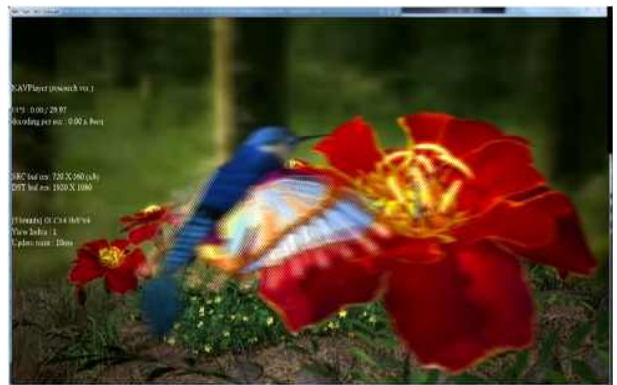


그림 14. 제안된 시스템의 실행 화면  
Fig. 14. Screen capture of the proposed system

그림 14는 제안하는 시스템 실행화면이다. 본 논문에서

는 복호화, 영상합성, 렌더링에 대하여 각각 8:3:1의 비율로 데이터 단위 병렬화 강도를 설정하였다.

### V. 시뮬레이션 결과

실험용 입력 압축영상 시퀀스로는 V3i사의 8시점 무안경식 입체 디스플레이에서 재생하기 위해 8개 시점으로 구성된 영상을 사용하였다. 다시점 압축영상이 복호화 된 후, 합성 알고리즘을 거쳐 렌더링 되고 화면에 표시되는 과정을 초당 프레임수로 계산하여 병렬화 단계별로 비교하였다. 세부 실험조건은 표 2와 같다.

4장에 도시된 그림 14는 8시점 합성 영상을 재생 중인 실행화면이다. 합성알고리즘이 올바르게 적용 되어 무안경식 다시점 입체 디스플레이에서 재생 시 입체감을 느낄 수 있음을 확인하였다. 그림 15는 2차원 모드 재생 시 동일한 시간위치에 대하여 8개의 시점을 전환해서 본 화면이다. 1(좌시점)부터 8(우시점) 까지 정지 또는 재생 중 자유로운 시점전환이 가능하다.

그림 16은 제안된 시스템의 병렬화에 따른 최대 프레임 처리속도 차이를 도시하였다. 다중코어 프로세서에서 작업 단위 병렬화 후 처리속도가 소폭 상승하였고, 추가적으로 데이터 단위 병렬화로 부하 균형 문제를 해결한 후 처리속

도가 더욱 상승하였다.

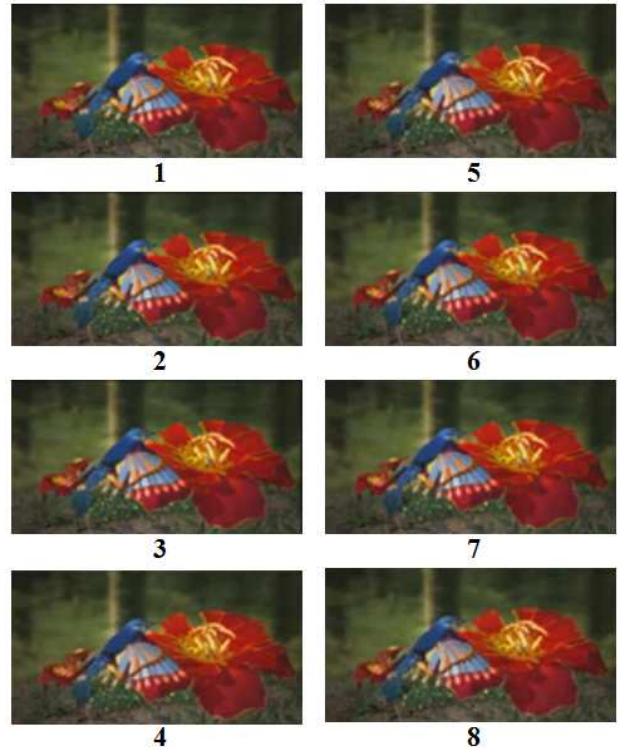




그림 15. 2차원 모드 재생 시 시점 전환  
Fig. 15. Viewpoint change on 2D mode

표 2. 시험조건  
Table 2. Test Condition.

해상도 및 프레임률	 Bird(720x360p, 30Hz) x 8 views	 Bottle(720x360p, 30Hz) x 8 views
프레임 수	900 x 8 frames	900 x 8 frames
압축 포맷	H.264/AVC High Profile	MPEG-2 Main Profile
프로세서	AMD Phenom 2.50GHz (4 core), Intel Xeon 3.0GHz (8 core)	
운영체제	Windows XP	

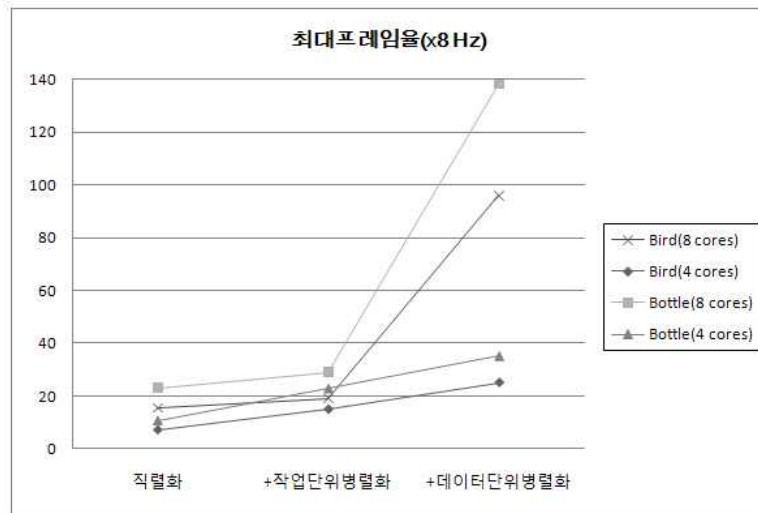


그림 16. 병렬화와 최대처리속도 (8 multi-frames/seconds)  
 Fig. 16. Maximum processing speed on the parallelization (8 multi-frames/seconds)

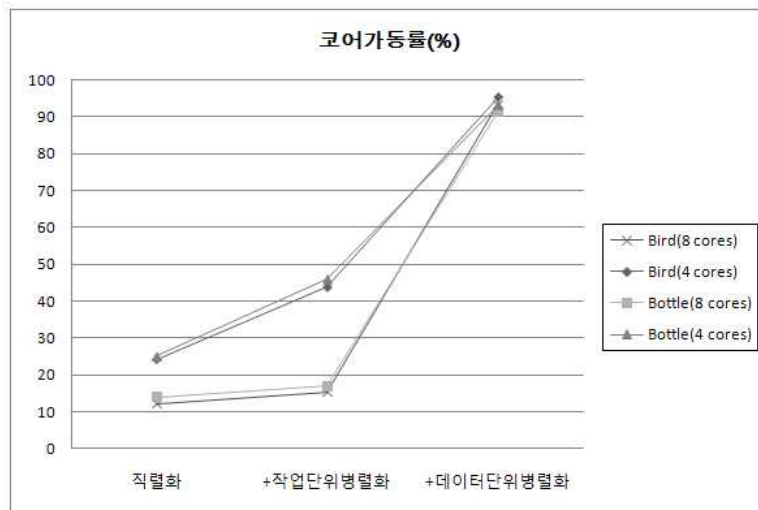


그림 17. 병렬화와 코어가동률  
 Fig. 17. Rate of core operation on the parallelization

그림 17은 병렬화에 따른 시스템 실행 시 전체 코어 가동률의 차이를, Intel Parallel Amplifier<sup>[5]</sup>로 측정된 측정치로 보여주고 있다. 두 측정값 모두 작업 단위 병렬화만 수행 시 부하균형이 맞지 않아서 최대프레임률 및 코어가동률에서 성능향상이 작지만, 데이터 단위 병렬화 후 master-worker 구조를 사용하여 부하균형을 맞추고 나서 성능

이 크게 향상되었다.

'Bird' 시퀀스에서 병렬화 이전에 비하여 병렬화 이후의 최대 프레임 처리율이 8개의 코어에서 6.19배, 4개의 코어에서 3.47배 향상되었다. 'Bottle' 시퀀스의 경우 최대 프레임 처리율이 8개의 코어에서 5.98배, 4개의 코어에서 3.32배 향상되었다. 또한 두 시퀀스 모두 병렬화 후 프로세서의

전체 코어 가동률을 각각 93%, 95% 까지 활용함으로써 시스템의 운영체제 구동에 관한 자원소모를 감안하였을 때 최대가동률에 근접한 결과를 얻었다. 이는 시스템이 실행 중 모든 코어에 부하가 고르게 분산되어 특정 코어만 유휴상태인 시간이 최소화되었음을 의미한다.

구현 시 복수개의 코어로 구성된 중앙처리장치(Central Processing Unit)를 사용하였는데 더 향상된 방법으로는 최근 프로그래밍 가능한 그래픽처리장치(Graphics Processing Unit)를 사용하여 반복되는 단일연산을 고속으로 병렬 수행하는 GPGPU(General-Purpose computing on Graphics Processing Units)기술을 사용할 수 있다. 그래픽처리장치는 중앙처리장치보다 각 코어의 처리속도는 느리지만 중앙처리장치보다 수십에서 수백 배의 훨씬 많은 수의 스트레드에서 동일한 계산을 병행처리가 가능하기 때문에 다시점 비디오 재생 시스템과 같이 대용량 데이터를 처리해야 할 때 유용하다. 따라서 작업 간 부하균형을 실시간으로 조절할 시 중앙처리장치를 사용하고 파이프라인의 각 단계에서 데이터 단위 병렬화 시 GPGPU 기술을 사용하여 많은 수의 코어에서 비트스트림 처리를 한다면 더욱 높은 속도향상을 기대할 수 있다<sup>[13]</sup>.

## VI. 결 론

본 논문에서는 기존 비디오 재생 시스템 보다 고속으로 동작하는 다시점 3차원 비디오 재생 시스템에 대한 연구를 수행하였다. 전체 처리속도 향상을 위해 다중 프로세서 환경에서 최적의 속도를 얻을 수 있는 작업 단위 및 데이터 단위 병렬화 방법을 제안하였다. 소프트웨어로 구현한 시스템을 시험한 결과 병렬화 이전에 비하여 병렬화 이후의 최대 프레임 처리율이 8개의 코어에서 약 6.2배, 4개의 코어에서 약 3.4배 향상되었으며 이는 코어 수 증가에 따른

이론적인 성능향상 최대치에 근접한 우수한 결과였다. 또한 각 코어에 부하가 고르게 분산 되어서 프로세서의 전체 코어 가동률이 기타 자원소모를 감안한 최대치인 약 94% 까지 상승 가능성을 확인하였다.

## 참 고 문 헌

- [1] C.V. Berkel, D.W. Parker, A.R. Franklin, "Multiview 3D LCD", Proc. SPIE, Vol. 2653, 32, pp. 32 - 39, January, 1996.
- [2] S.S. Kim, B.H. You, H. Choi, B.H. Berkeley, D.G. Kim, N.D. Kim, "World's First 240Hz TFT-LCD Technology for Full-HD LCD-TV and Its Application to 3D Display", SID 09 DIGEST, pp. 424-427, September, 2009.
- [3] K. Sakamoto, R. Kimura, M. Takaki, "Parallax Polarizer Barrier Stereoscopic 3D Display Systems", Active Media Technology, pp. 469 - 474, May, 2005.
- [4] W. Knight, "Two Heads are Better than One", IEEE Review, pp 32-35, September 2005.
- [5] 이용범, 정세일, "패럴랙스 배리어 및 이를 구비한 입체영상 디스플레이 장치 (Multi-view 3D display apparatus and method)", 대한민국 등록특허 (KR) 10-0938481-0000, (2010.01.15).
- [6] N.A. Dodgson, "Analysis of the viewing zone of the Cambridge autostereoscopic display", Applied Optics, Vol. 35, no. 10, pp. 1705-1710, April 1996.
- [7] 이승현, "패럴랙스배리어와 렌티큘러 스크린을 이용한 3D 디스플레이", 인포메이션디스플레이, 10권, 3호, pp. 15-23, 2009년 6월.
- [8] Intel Parallel Amplifier (<http://software.intel.com/en-us/intel-parallel-amplifier/>).
- [9] G.M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities", Spring joint computer conference, Atlantic City, New Jersey, April, 1967.
- [10] T.G. Mattson, B.A. Sanders, B.L. Massingill, Patterns for Parallel Programming, Addison-Wesley Professional, 2004.
- [11] N. Carrier, D. Gelernter, How to write parallel programs - A first course, MIT Press, Cambridge, MA, 1990.
- [12] S.G. Gagne, Operating system concepts, 6th edition, John Wiley & Sons, Inc, 2003.
- [13] O. Schenk, M. Christen, H. Burkhart, "Algorithmic performance studies on graphics processing units", Journal of Parallel and Distributed Computing 68, pp. 1360 - 1369, October, 2008.

---

저 자 소 개



허 영 수

- 2010년 2월 : 경희대학교 컴퓨터공학과 학사
- 2010년 3월 ~ 현재 : 경희대학교 컴퓨터공학과 석사과정
- 주관심분야 : 비디오 신호처리, 멀티미디어, 영상처리, 고성능컴퓨팅



박 광 훈

- 1985년 2월 : 연세대학교 전자공학과 학사
- 1987년 7월 : 연세대학교 전자공학과 석사
- 1991년 1월 : Case Western Reserve University, Dept. of EEAP 석사
- 1995년 1월 : Case Western Reserve University, Dept. of EEAP 박사
- 1995년 3월 ~ 1997년 2월 : 현대전자 멀티미디어연구소 책임연구원
- 1997년 3월 ~ 2001년 2월 : 연세대학교 문리대학 전산학과 부교수
- 2001년 3월 ~ 현재 : 경희대학교 컴퓨터공학과 교수
- 주관심분야 : 멀티미디어, 비디오 신호처리, 패턴인식, 영상처리, 계산지능