

멀티미디어 프리젠테이션 시스템을 위한 보정적 동기화 알고리즘

정희경
배재대학교 전자계산학과

A Compensative Synchronization Algorithm for Multimedia Presentation System

H.K. Jung
Department of Computer Science, PaiChai University

본 논문은 멀티미디어 프리젠테이션을 위한 보정적 동기화 알고리즘에 관한 연구이다. 멀티미디어 프리젠테이션 시스템에서 표현되는 미디어들의 지연시간을 미리 예측하여 보정해 줌으로써 지연시간을 최소화하는 보정적 동기화 알고리즘을 제안하였다. 이 보정적 동기화 알고리즘을 이용하여 텍스트, 라스터 그래픽스, 도형 그래픽스, 오디오와 같은 미디어 객체를 갖는 멀티미디어 프리젠테이션 시스템을 설계 구현하였다.

This paper describes the design and implementation of multimedia presentation system based on the compensative synchronization algorithm. We proposed the compensative synchronization algorithm in compensation for previously preestimated delay-time in order to minimize delay-time in multimedia presentation system. The presentation system uses compensative synchronization algorithm and have multimedia objects as text, raster graphics, geometric graphics, or audio.

Key words : synchronization, Multimedia, presentation system, compensative, algorithm.

I. 서 론

컴퓨터의 하드웨어와 소프트웨어 기술의 급속한 발전에 따라 정보 표현 미디어는 텍스트, 도형 그래픽스, 라스터 그래픽스 등의 정적 미디어 뿐만 아니라 애니메이션, 비디오, 오디오 등과 같은 시간에 의존한 특성을 갖는 동적 미디어 표현이 요구되고 있다. 또한 효율적인 정보 전달을 위해 이러한 미디어들을 통합하여 표현하기 위한 멀티미디어 프리젠테이션 시스템에 대해 많은 연구가 되고 있으며, 특히 시간 동기화 문제를 해결하려는 연구가 요구되고 있다.¹⁻⁷⁾

기존의 동기화에 대한 연구에서 멀티미디어 데이터들을 저장매체에서 인출하여 표현 매체 상에서 수행되기까지의 시간인 준비시간 문제가 여러

움의 요인이 되었다. 준비시간은 데이터 전송시간, 데이터의 인코딩 및 디코딩 시간, 네트워크 지연시간, 스케줄링 시간, 동기화 구조 내부의 지연시간 등으로 구성된다. 이러한 준비시간 문제점을 해결하기 위한 노력이 모니터/스케줄러 프로세서를 이용한 방식으로 준비 시간 문제를 해결하기 위해 Scripted 선인출 기법을 적용하는 방식을 사용한다. 그리고 각 미디어를 채널로 분리하여 독립적으로 수행시켜 최대 준비시간을 이용하여 선인출의 시작은 Pre-arming 이벤트(event)로서 전달하는 방식으로 미디어 데이터의 종류, 크기, 네트워크 환경에서의 데이터 위치 등을 고려하여 이벤트를 설정한다. 각각의 이벤트는 채널에 의해 구분되어 서로 간에 독립적으로 수행되며 이들 간의 구성 방식에 대한 고려 사항을 언급한

다. 또한 최대 준비시간을 사용한 최적 선인출 시기를 결정하는 알고리즘이 제안되었다.^{8,9)}

그러나 이들은 구체적인 선인출 시기 설정에 대한 언급 및 방법에 대한 설명이 부족하고, 시스템 자원의 효율은 고려하고 있지 않으며, 동기화에 따른 프리젠테이션 시스템 설계 및 구현에 대한 연구는 부족한 실정이다.

이에 본 논문에서는 표현 미디어 객체들의 크기에 따른 시작 및 실행 지연시간을 미리 실험을 통해 예측하여 시작 지연시간과 실행 지연시간을 보정하여 지연시간을 최소화하는 보정적 알고리즘을 제안하였다. 또한 이 알고리즘을 이용하여 텍스트, 도형 그래픽스, 라스터 그래픽스, 오디오 등을 표현 가능한 멀티미디어 프리젠테이션 시스템을 설계 구현하였다.

구현 환경은 IBM-PC 환경 하에서 단일 사용자 환경에서 구현하므로 준비시간에서 네트워크 지연시간을 무시하였고, 이식성이 좋고 사용자에게 친숙하고 일관된 사용자 접속을 제공하는 MS-Windows에서 구현하였다.¹⁰⁾

논문의 구성은 제 2 장에서 보정적 동기화 알고리즘을 설명하고, 제 3 장에는 알고리즘을 이용한 멀티미디어 프리젠테이션 시스템에서의 실험 및 고찰을 설명한다. 제 4 장에는 결론에 대해 기술한다.

II. 보정적 동기화 알고리즘

일반적으로 시간 동기 표현방식은 두 가지가 있다. 첫째 방식은 Allen¹¹⁾의 13가지 시간관계 모델을 표현하기에 적합하지만 한 미디어의 정보 손실로 전체에 영향을 미친다. 둘째는 절대 시간 출력 상에서 절대적 시간 관계 표현방식으로 작성된 동기 정보 시스템에서 쉽게 이용가능하며 시간관계를 자세히 규정할 수 있으나 사용자가 직접 작성하기가 어려운 단점이 있다.^{12,13)} 본 논문은 후자의 방식으로 단일 사용자 환경에서 구현하여 준비시간에서 네트워크 지연시간을 제외한 나머지를 보정적 동기화 알고리즘으로 개선할 수 있다. 보정적 동기화 구조는 직렬 동기화와 병렬 동기화로 나누어 보정을 한다.

1. MHEG의 동기화 표현 방식^{5,6)}

MHEG은 멀티미디어/하이퍼미디어의 정의, 코

딩 원칙 및 시스템 요구사항과 MHEG이 정의하고 있는 멀티미디어 객체 클래스(class)를 이용한 멀티미디어/하이퍼미디어 정보의 표현을 목적으로 ISO/IEC JTC1/SC29에서 제정한 표준이다.

가. MHEG 객체 정의

MHEG은 객체 지향 방식을 사용하여 객체로 표현되는 여러 가지 데이터 유형을 정의한다. 단일 객체로 텍스트, 이미지, 비디오, 오디오 등과 같은 한가지의 특정 정보를 포함하고 있는 객체이고, 단일 객체들이 혼합된 형태를 복합 객체라 한다. 복합 객체내의 각 객체들은 지정된 상호관계를 가지고 동작한다. MHEG 객체는 입력(input), 출력(output), 대화(interactive), 하이퍼객체(hyperobject)의 네 가지 유형이 있다.

나. MHEG 동기화 모델

객체간의 동기화 관계는 시간과 공간축상에서 나타낼 수 있으며, 멀티미디어 시스템에 필요한 전체적인 동기화 기능은 다음과 같이 7가지의 동기화 방식으로 구분하여 설명할 수 있다.

1) 원형 병렬 동기화

(Atomic Parallel Synchronization)

객체들이 동시에 시작하여 병렬적으로 표현되도록 한다. 이는 객체의 시작 이벤트에 동기화 되는 것으로 구현할 수 있다.

2) 원형 직렬 동기화

(Atomic Serial Synchronization)

하나의 객체가 끝나고 다음 객체가 연속적으로 연결되어 동작하도록 한다. 이는 객체의 종료 이벤트에 동기화 되는 것으로 구현가능하다.

3) 기본형 병렬 동기화

(Elementary Parallel Synchronization)

시작시간의 지연시간이 서로 다르면서 병렬적으로 객체를 동작하도록 한다.

4) 기본형 직렬 동기화

(Elementary Serial Synchronization)

객체의 지연시간을 갖고 연속적으로 객체들을 표현하도록 한다.

5) 반복형 동기화(Cyclic Synchronization)

객체의 반복적인 동작을 하도록 한다.

6) 조건형 동기화(Conditional Synchronization)

특정 이벤트나 응답에 따라 해당하는 객체를 표현한다.

7) 사슬형 동기화(Chained Synchronization)

단일 객체들을 연결하여 새로운 복합 객체로

표현하거나 다른 객체와 동기화 되도록 한다.

2. 동기화

MPEG 동기화를 지원하는 동기화는 다음과 같다. 직렬 동기화는 한 미디어 객체의 종료시간이 다른 미디어 객체의 시작시간에 영향을 주는 동기화 방식이다. 병렬 동기화는 미디어 객체를 표현할 때 한 미디어 객체의 종료시간이 다른 미디어 객체의 시작시간에 영향을 미치지 않고 단지 미디어 객체의 표현에서 발생하는 시작 지연시간에만 영향을 주는 동기방식으로 이들 간의 관계를 그림 1에 보인다.

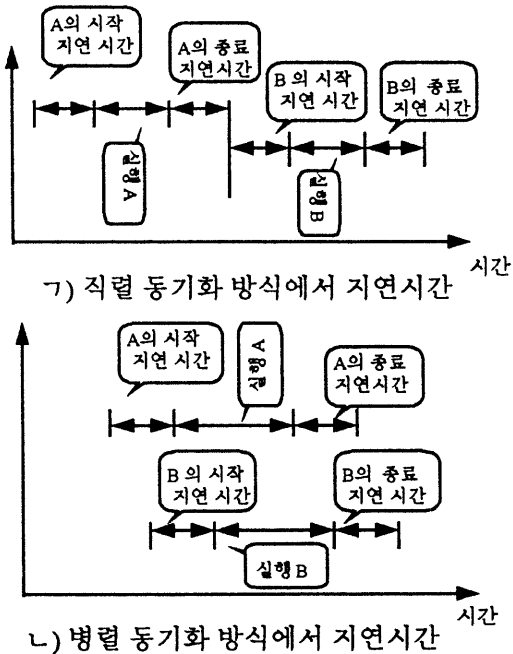


Fig.1 Delay-time in Serial/Parallel Synchronize Methods.

직렬에서 시작 지연시간과 종료 지연시간은 각각 시작 지연시간의 보정과 다음에 수행될 미디어 객체의 시작시간을 위해 보정처리 해야 하는 지연시간이다. 그러나 병렬에서는 한 미디어 객체의 종료시간이 다른 미디어 객체 시작시간에 영향을 미치지 않으므로 단지 미디어 객체의 시작 지연시간이 미디어를 표현 시에 영향을 준다. 실제 미디어 객체가 지정된 시작시간에 시작되고 종료되기 위한 시간 계산은 다음과 같다.

미디어 객체 A의 시작시간

- = 지정된 시작시간 - A의 시작 지연시간;
- 미디어 객체 A의 종료시간
- = 지정된 종료시간 - A의 종료 지연시간;
- 미디어 객체 B의 시작시간
- = 미디어 객체 A의 종료시간 - B의 시작 지연시간;
- 미디어 객체 B의 종료시간
- = 지정된 종료시간 - B의 시작 지연시간;

3. 보정적 동기화 처리

보정적 동기화 처리는 멀티미디어 데이터 표현에서 나타나는 지연시간을 해결하기 위해 실행할 미디어 객체의 지연시간을 계속 측정하여 나타나는 지연시간을 보정하여 사용자가 설정한 시간에 미디어 객체가 정확히 표현되도록 하는 처리를 말한다.

각 지연시간을 해결하기 위한 보정적 동기화 알고리즘을 그림 2에 기술한다.

```

Compensative_sync_algorithms
{
    Set_time(S_time, E_time);
    // 각 미디어에 대한 시작시간과
    // 종료시간을 지정
    while(event)
    {
        if (최초 실행)
            고정 지연시간 설정;
            // 시작 지연시간에 대한 실험
            // 데이터 화일 참조
        지연시간 보정처리;
        if (시스템의 현재시간 == 미디어 객체 시작시간)
        {
            각 미디어 처리별로 미디어 객체 표현 출력;
            // 오디오, 애니메이션, 텍스트, 도형
            // 그래픽스, 라스터 그래픽스 객체 표현
            각 미디어 객체 실행시 실제 지연시간 측정;
        }
    }
}
    
```

Fig.2 Algorithm of compensative synchronization.

프리젠테이션 시스템의 시간선(time line) 편집기에 의해 시작과 종료시간이 지정되고, 최초 실행 시에 많은 표현 미디어 객체가 존재할 때 많이 지연되므로 미리 실험을 통해 각 미디어 객체에 대한 지연시간 데이터 화일을 가지고 비례적

으로 계산함으로써 초기의 지연시간을 많이 줄일 수 있다. 그리고 지연시간을 보정한 뒤 현재 시스템 시간과 미디어 객체의 시작시간을 비교하여 일치하면 해당 미디어 객체의 이벤트를 발생한 뒤 각 미디어 객체의 실행 시에 발생하는 실제 지연시간을 측정하여 계속하여 지연시간 보정을 함으로써 실제 사용자가 원하는 요구에 따르도록 각 미디어 객체들이 표현될 수 있도록 하였다.

4. 멀티미디어 프리젠테이션 시스템

일반적으로 멀티미디어 프리젠테이션 시스템에서 효율적인 정보 표현을 위해 다음과 같은 조건을 만족해야 한다.

첫째, 정적 미디어뿐만 아니라 동적 미디어를 함께 표현할 수 있어야 한다.¹⁴⁾ 둘째, 미디어들을 다양한 형태로 표현 가능해야 한다.¹⁵⁾ 셋째, 각 미디어 객체들 간의 미디어 표현에서 동기화 되어야 한다.¹⁴⁾ 넷째는 다양한 미디어를 포함하여 비순차적으로 저작할 수 있는 저작기능을 가져야 한다.¹⁶⁾ 본 논문에서는 이러한 조건을 만족하도록 하기 위해 보정적 동기화 알고리즘을 바탕으로 하여, 미디어 객체들은 텍스트, 도형 그래픽스, 오디오, 애니메이션 등의 미디어를 포함하며, 텍스트 등이 움직이거나 배경 화상을 다양하게 변화하도록 하며, 보정적 동기화 기법에 의해 동기화 처리를 한다. 또한 텍스트, 도형 그래픽스 등을 편집하고 원하는 시간에 또는 이벤트가 발생하도록 하는 저작 기능을 갖는 프리젠테이션 시스템을 설계 구현한다. 프리젠테이션 시스템 구성은 그림 3과 같다.

시스템은 사용자 인터페이스, 동기화를 위한 시간선 편집기, 미디어의 편집을 위한 미디어 편집기 및 동기화 보정을 위한 보정적 동기화 처리 모듈로 구성된다. 사용자 인터페이스는 사용자가 아이콘을 이용하고, 마우스를 사용하여 미디어 편집기와 시간선 편집기를 호출할 수 있으며 화일을 열거나 저장할 수 있다. 시간선 편집기는 사용자가 동기화될 텍스트, 도형 그래픽스, 라스터 그래픽스, 오디오, 애니메이션 화일의 화일명, 시작시간, 종료시간, 표시될 크기와 위치를 입력 받아 원하는 시간과 공간에 표현되도록 편집할 수 있다. 편집된 동기화 화일은 지정된 시간을 보정적 동기화 처리하여 원하는 시간에 사용자에게 출력시켜 준다. 미디어 편집기는 텍스트 화일을 생성하고, 간단한 도형 즉, 원, 선, 사각형 등의 도형 그래픽스 화일을 생성하며, 윈도우즈에서 제공하는 화일(BMP 포맷)을 축소/확대하여 라스터 그래픽스 화일을 만든다. 또한 오디오 화일은 사운드 브라스터 호환 화일을 재생할 수 있으며, 애니메이션 화일을 3D Studio에서 제공하는 화일(FLI 포맷)을 재생할 수 있도록 제공한다.^{10,17,18)}

III. 실험 및 고찰

1. 구현

본 시스템의 구현 환경은 IBM-PC 586 호환기종에서, 사용자 상호접속을 위한 윈도우 시스템으로는 MS-Windows 3.1을 사용하였으며, 윈도우즈에서의 멀티미디어 타이머 인터럽트를 사용하여 동기화를 조정하였다. 출력장치는 포스트스크립트(PostScript)가 내장된 프린터로 출력하였다. RAM은 16Mbyte, 하드 디스크는 540Mbyte, 그래픽 카드는 800 x 600의 256 컬러를 지원하는 SuperVGA 그래픽 카드를 사용하였으며, 입력을 위해서는 키보드와 마우스를 사용하였다. 오디오를 위해서는 사운드 카드로 사운드 블래스터 프로를 사용하였으며, 애니메이션 화일을 위해서는 3D Studio 응용 프로그램을 이용하였다. 라스터 그래픽스 화일은 300 dpi의 이미지 스캐너를 이용하여 데이터 화일을 생성하였다.

2. 실험

최초 실행시 고정 지연시간 설정을 위해 오디

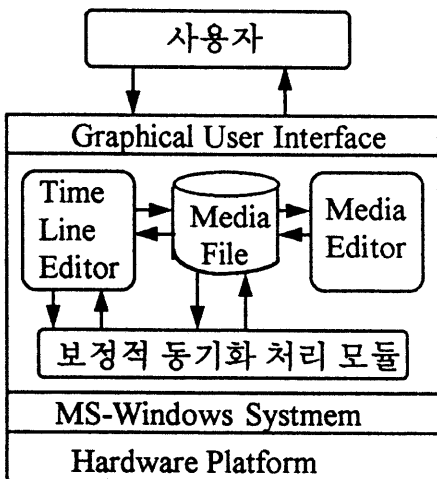


Fig. 3 Diagram of Presentation System.

오, 애니메이션, 라스터 그래픽스, 텍스트와 같은 미디어 객체 데이터에 대해 실험을 반복하여 실험 데이터의 고정 시작시간을 정하였다. 각 미디어 객체 데이터에 대해 데이터의 크기를 변경하며 3회의 실험을 통해 얻은 데이터의 시작 지연시간 그래프를 그림 4, 그림 5, 그림 6, 그림 7에 각각 보인다.

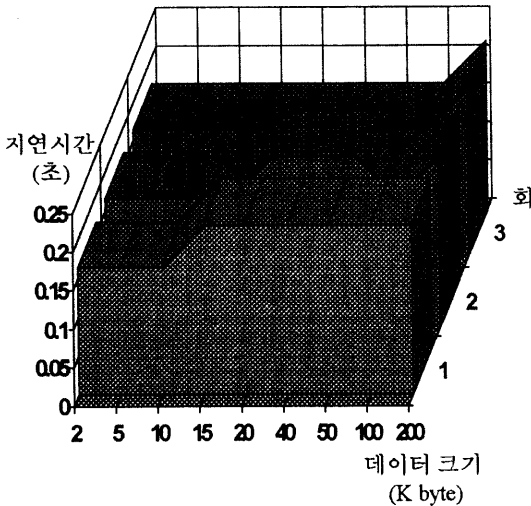


Fig. 4 Start Delay-time Graph of Audio Data.

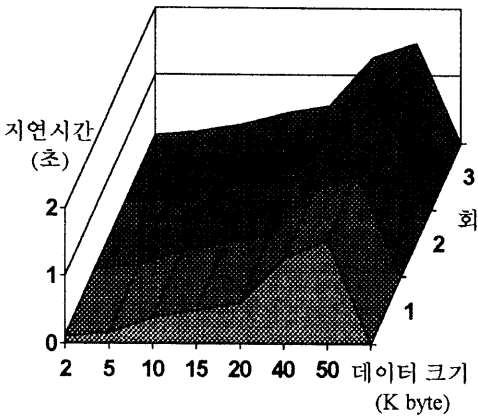


Fig. 5 Start Delay-time Graph of Animation Data.

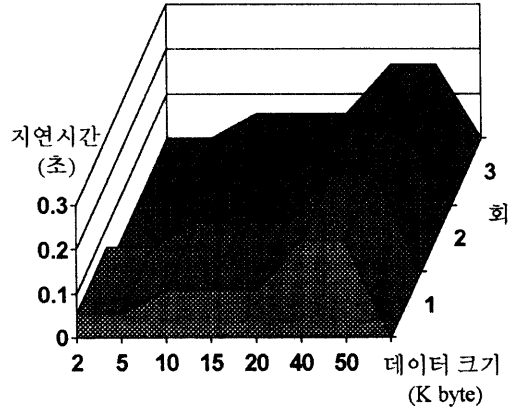


Fig. 6 Start Delay-time Graph of Raster Graphics

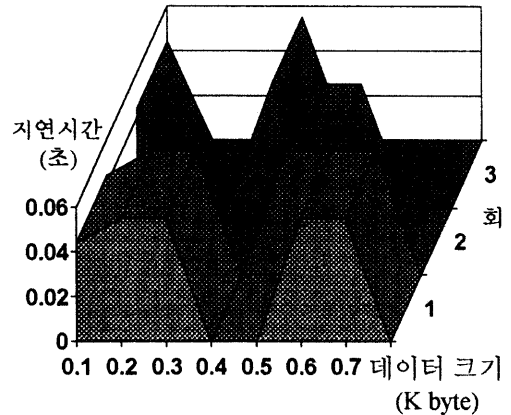


Fig. 7 Start Delay-time Graph of Text Data.

그래프에서처럼 오디오 데이터의 크기에 따른 시간의 변화가 완만한 것으로 크기에 크게 영향을 받지 않음을 알 수 있으며, 애니메이션 데이터에서는 데이터의 크기가 커짐에 따라 시작 지연시간이 증가함을 볼 수 있다. 또한 라스터 그래픽스 데이터는 데이터의 크기가 커짐에 따라 시작 지연시간이 증가함을 알 수 있으며, 텍스트 데이터는 크기에 크게 영향을 받지 않음을 알 수 있다.

그림 8은 임의의 각 미디어 객체 데이터에 대해 임의의 시간 후에 동시에 이벤트가 발생하도록 할 때 보정적 동기화 기법을 적용한 경우와 하지 않은 경우를 3회 실시한 후의 데이터에 대한 그래프이다. 보정적 동기화 방식을 적용하였을 때 현저히 지연시간의 감소를 확인할 수가 있다. 실험 회수를 증가할수록 지연시간의 값이 보정되어 지연시간이 감소함을 확인할 수가 있다.

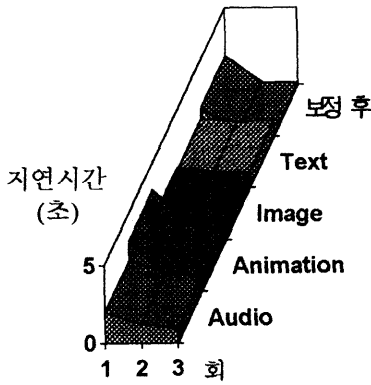


Fig.8. Delay-time According to Compensative Synchronization Algorithms.

그림 9는 본 프리젠테이션 시스템으로 애니메이션, 텍스트, 라스터 그래픽스, 오디오 데이터로 배경음악을 갖는 화면 구성 예를 보인다.



Fig. 9. Example of Presentation.

3. 고 찰

본 논문은 보정적 동기화 알고리즘을 제안하여 멀티미디어 프리젠테이션 시스템을 구현하였다. 논문에서 제시한 보정적 동기화 알고리즘은 멀티미디어 데이터 표현에서 나타나는 지연시간을 해결하기 위해 기 실행된 결과를 가지고 지연시간의 값을 보정해 줌으로써 많은 감소 효과를 가져왔음을 확인할 수 있었으며, 하드웨어에 독립적으로 실현할 수 있었다. 그러나 하드웨어 인터럽트 특성상 각 미디어 객체에 대해 0.055초 이하로 시작시간을 지정할 수 없었고, 데이터 크기가

큰 경우는 데이터 전송시간 때문에 많은 지연을 가져왔다. 이에 대한 효율적인 처리를 위해 동적 버퍼 분할과 더블 버퍼링에 대한 연구가 요구된다.

앞으로 MHEG 표준에서 제안하고 있는 동기화를 모두 포함하고, 단일 사용자 환경뿐만 아니라 다중 사용자 환경에서도 쉽게 적용될 수 있도록 준비시간에 포함된 네트워크 지연시간에 대한 연구도 요구된다.

본 시스템은 일반적으로 멀티미디어 프리젠테이션 시스템에서 요구되는 조건으로 정적미디어뿐만 아니라 동적 미디어를 포함하고, 미디어들을 다양하게 표현하며, 표현 미디어들의 동기화 및 비순차적으로 저작기능을 가져야 하는 조건을 만족함으로써 멀티미디어 저작도구 개발과 교육용이나, 홍보용 등의 정보전달용 시스템에 크게 기여하리라 본다.

IV. 결 론

근래 컴퓨터의 하드웨어와 소프트웨어의 발전에 따라 다양한 미디어를 혼재하여 표현하고자 요구되고 있다. 이에 본 논문은 실험 데이터를 이용하여 미디어 표현시 지연시간을 줄일 수 있도록 보정적 동기화 알고리즘을 제안하였다. 이 제안된 보정적 동기화 알고리즘을 이용하여 텍스트, 도형 그래픽스, 라스터 그래픽스와 같은 정적 미디어뿐만 아니라, 오디오와 애니메이션과 같은 동적 미디어를 포함하고, 사용자가 원하는 시공간에 미디어 객체 표현이 가능한 저작기능을 갖는 멀티미디어 프리젠테이션 시스템을 설계 및 구현하였다. 시스템은 하드웨어에 독립적으로 실현할 수 있었으며, 시간지연을 줄일 수 있었다.

본 시스템은 일반적인 멀티미디어 프리젠테이션 시스템에서 요구되는 조건들을 충족함으로써 저작도구 개발과 교육용, 홍보용 등의 응용에 크게 기여하리라 본다.

이후 연구 방향으로는 MHEG 표준에서의 동기화를 모두 포함하도록 확장하고, 다중 사용자 환경에서도 적용될 수 있도록 네트워크 지연시간에 대한 연구가 요구되며, 새로운 미디어의 지원과 다양한 표현의 프리젠테이션 시스템에 대한 보다 많은 연구가 되어야 할 것이다.

감사의 말씀

본 논문은 95년도 배재대학교 교내학술연구비 지원에 의하여 수행된 연구의 일부로 이에 감사를 드립니다.

참고 문헌

1. P. Hoepnor., "Synchronizing the Presentation of Multimedia Object-ODA Extension," Eurographics Multimedia Workshop, Stockholm, pp. 19~32, Apr. 1991.
2. T. D. C. Little, et al., "Spatial-temporal Composition of Distributed Multimedia Objects for Value-Added Networks," pp. 42~50, Oct. 1991.
3. R. Steinmetz., "Multimedia Synchronization Techniques: Experiences Based on Different System Architectures," 4th IEEE ComSoc International Workshop on Multimedia Communications, pp. 306~314, Apr. 1992.
4. Kazutoshi Fujikawa, et al., "Multimedia Presentation System Harmony with Temporal and Active Media," Proceedings of USENIX-Summer'91, pp 75~93, 1991.
5. Information Technology-Coded Representation of Multimedia AND Hypermedia Information Objects(MHEG), ISO CD13522-1.
6. Brian D. Markey, et al, "HyTime and MHEG," COMPCON SPRING, pp. 25~40, Feb. 1992.
7. T. Sato, et al, "A Multimedia Synchronization Model using Logic Operators," Processing of 7th International Joint Workshop on Computer Communication, pp. 69~76, Jun. 1992.
8. 송정적, et al., "버퍼 효율을 고려한 멀티미디어 데이터 선인출 기법," 1994년도 한국정보과학회 가을 학술발표논문집 Vol. 21, No. 2, pp. 43~46, 1994.
9. Guido van Rossum, et al, "CMIFed : A Presentation Environment for Portable Hypermedia Document," '93 ACM Multimedia, pp. 183~188, Jun. 1993.
10. Microsoft Corporation, "Multimedia Programmer's Guide," 1992.
11. Allen, "Maintaining Knowledge about Temporal Interval," Comm. of the ACM, Vol. 26, No. 11, Nov. 1983.
12. Seong Bae Eunn, et.al, "Specification of Multimedia Composition and a Visual Programming Environment," ACM Multimedia 93, Anaheim, California, pp. 167~173, Aug. 1993.
13. Simon Gibbs, et. al, "An Object-oriented Framework for Multimedia composition and Synchronization," Eurographics Multimedia Workshop, Stockholm, pp. 133~147, Apr. 1991.
14. Susan Angelbrandt, et al., "Integrating Audio and Telephony in a Distributed Workstation Environment," Proceedings of USENIX-Summer'91, pp 419~435, 1991.
15. Polle T. Zellweger, "Scripted Documents: A Hypermedia Path Mechanism," Proceedings of Hypertext'89, pp. 1~14, Nov. 1989.
16. Barry Arons, "Hyperspeech : Navigating in Speech-only Hypermedia," Proceedings of Hypertext'91, pp. 136~146, 1991.
17. Charles Petzold, "Programming Windows 3.1," 1992.
18. Creative Lab., "Sound Blaster Pro Reference Manual," 1994.